



Universidad Autónoma de San Luis Potosí
Facultad de Ingeniería
Centro de Investigación y Estudios de Posgrado (CIEP)

Sistema de asistencia para el diagnóstico de enfermedades cardiovasculares basado en procesamiento de señales electrocardiográficas y árboles de decisión

Tesis presentada como requisito
para obtener el grado de
Maestro en Ingeniería de la Computación

Autor:

Alberto Jorge Navarro Ramírez

Director:

Dr. Carlos Soubervielle Montalvo

Co-director:

Dr. César Augusto Puente Montejano

San Luis Potosí, S.L.P.
Febrero 2026

Dedicatoria

*A mi familia,
por su amor incondicional, su paciencia infinita
y por enseñarme con su ejemplo que ningún sueño
está fuera de nuestro alcance.*



Sistema de asistencia para el diagnóstico de enfermedades cardiovasculares
basado en procesamiento de señales electrocardiográficas y árboles de decisión
© 2026

by Alberto Jorge Navarro Ramírez

is licensed under CC BY-NC-ND 4.0.

<https://creativecommons.org/licenses/by-nc-nd/4.0/>

Agradecimientos

Este trabajo es el resultado de muchas manos, muchas horas y muchas voluntades. A todas ellas, mi gratitud.

Al **Dr. Carlos Soubervielle Montalvo** y al **Dr. César Augusto Puente Montejano**, director y co-director de esta tesis, por su guía atenta, su rigor académico y su confianza en la propuesta. Cada observación y cada reunión marcó un avance concreto en la calidad de este proyecto.

Al **Dr. Juan Manuel López Quijano**, **Cardiólogo y Subdirector de Educación e Investigación del Hospital Regional de Alta Especialidad “Dr. Ignacio Morones Prieto”**, por revisar con dedicación numerosos registros, por sus lecturas clínicas de los electrocardiogramas y por el acompañamiento cercano que dio forma a la validación médica del sistema. Su compromiso con la precisión y la seguridad del paciente permea cada capítulo.

A la **Universidad Autónoma de San Luis Potosí**, por el respaldo institucional y el acceso a recursos técnicos que hicieron viable la implementación y verificación del sistema.

Al **Dr. Emilio Jorge González Galván**, **Director de la Facultad de Ingeniería**, por el apoyo brindado a mi formación y por impulsar condiciones que permiten que proyectos como este se desarrollen y concluyan con seriedad académica.

A mis **sinodales**, por su tiempo, su lectura cuidadosa y sus observaciones para fortalecer este trabajo:

- A la **Dra. Marcela Mejía Carlos**, por sus aportaciones y comentarios que ayudaron a mejorar la claridad y el sustento académico del documento.
- Al **Dr. Alberto Salvador Núñez Varela**, por su análisis crítico y recomendaciones para elevar la calidad metodológica y la presentación de resultados.
- A la **Dra. Sandra Edith Nava Muñoz**, por su revisión, sugerencias y acompañamiento en el proceso de evaluación.

A la **Dra. Claudia Elena González Acevedo**, Jefa de la Secretaría de Servicios Escolares de la Universidad Autónoma de San Luis Potosí, por su generoso apoyo y por facilitar los recursos indispensables que hicieron posible el desarrollo de esta investigación. Su respaldo oportuno abrió caminos cuando más se necesitaba.

Al **Mtro. Francisco Javier Martínez Jiménez**, Jefe del *Departamento de Admisiones, Control Escolar y Estadística*, por su comprensión y flexibilidad para armonizar mis responsabilidades laborales con los hitos de investigación. Su apoyo, como jefe directo, permitió que este trabajo avanzara sin perder el rumbo.

A todas las personas que aceptaron participar y a los pacientes del Hospital Regional de Alta Especialidad “Dr. Ignacio Morones Prieto”, por su tiempo y confianza. Su participación hizo posible el conjunto de datos y las evaluaciones que sostienen los resultados aquí presentados.

A mis amigos y compañeros de camino, por la paciencia, las conversaciones honestas y el ánimo cuando el cansancio apretaba. Gracias por estar, por empujarme a hacer las cosas bien y por recordarme que la vida también se construye con comunidad.

A todas aquellas personas que, desde su oficio, su familia, su trabajo y su servicio, **hacen de la sociedad un lugar mejor y luchan por un San Luis Potosí más justo y más humano**. Este proyecto también se sostiene en esa esperanza cotidiana: la de trabajar con verdad, con responsabilidad y con sentido de bien común.

A mi familia:

- A mi esposa, **Gabriela Irene Rodríguez Gámez**, por sostener el hogar, alentar cada intento y celebrar cada pequeño logro cuando el cansancio apretaba.
- A mis hijas, **María Pía Navarro Rodríguez** y **María Catalina Navarro Rodríguez**, por recordarme el para qué de este camino y convertir cada jornada en una razón para hacerlo mejor.
- A mi padre, **Leopoldo Navarro Ruíz**, y a mi madre, **María Araceli Ramírez Montenegro**, por la disciplina, el sentido del deber y la fe en el esfuerzo bien hecho.
- A mis hermanas, **María Daniela Navarro Ramírez** y **María Guadalupe Navarro Ramírez**, por su presencia constante y su apoyo en los momentos más demandantes.

Finalmente, y por encima de todo, a **Dios** y a **María Reina**, por sostener mi vida, mi vocación y mis decisiones, incluso en los días inciertos. Y a los laicos que han dado testimonio valiente de la fe y de la verdad, como el **Beato Anacleto González Flores**, y tantos otros que han elegido la vida, la dignidad y la coherencia como camino. Que este trabajo sea, en lo que le corresponde, un aporte honesto al servicio de las personas y al cuidado del corazón humano.

Resumen

En esta tesis se presenta el diseño, implementación y validación de un sistema de diagnóstico cardiovascular asistido que combina hardware de bajo costo (OpenBCI Ganglion y computadoras portátiles convencionales) con software ligero basado en Python, BrainFlow, PHP y MySQL. El flujo completo integra la adquisición de señal ECG en tiempo real, el filtrado digital, la extracción de once atributos clínicos esenciales y la clasificación mediante un árbol de decisión (*CART*) optimizado, entrenado y calibrado para emitir un diagnóstico preliminar *NORMAL* o *ANORMAL* que el médico puede revisar, así como la visualización interactiva vía web.

Los resultados en laboratorio controlado y con voluntarios sanos mostraron una detección de picos R con más del 97 % de exactitud y un error medio en intervalos RR inferior a 5 ms. En el conjunto de 90 registros etiquetados (45 normales y 45 anormales) utilizado para entrenamiento y validación cruzada en 8 pliegues, el clasificador *CART* alcanzó una exactitud media del 84.44 %, un F1-macro del 84.38 %, un *recall* del 77.78 % para la clase normal y del 91.11 % para la clase anormal, así como un AUC-ROC del 85.83 %. El sistema web, desarrollado con Bootstrap y Plotly, facilita la revisión remota de trazas ECG y la exportación de reportes en PDF.

Este trabajo demuestra que es posible ofrecer un apoyo diagnóstico preliminar fiable, de baja latencia y costo reducido (hardware menor a 500 USD, hospedaje menor a 5 USD/mes), apto para entornos de atención primaria y telemedicina en zonas de conectividad limitada. Se discuten fortalezas, limitaciones y líneas de trabajo futuro, entre ellas el soporte multicanal, el uso de modelos híbridos y la corrección avanzada de artefactos por movimiento. El énfasis en algoritmos interpretables y en un flujo completo —de la adquisición a la visualización— busca facilitar la adopción clínica y la colaboración entre ingeniería y medicina.

Abstract

This master’s thesis presents the design, implementation, and validation of an assisted cardiovascular diagnostic system that combines low-cost hardware (OpenBCI Ganglion and standard laptops) with lightweight software built on Python, BrainFlow, PHP, and MySQL. The end-to-end pipeline encompasses real-time ECG acquisition, digital filtering, extraction of eleven essential clinical features, and classification using an optimized decision tree (*CART*) trained and calibrated to provide a preliminary *NORMAL* vs. *ABNORMAL* diagnosis that can be reviewed by the physician, together with an interactive web-based visualization module.

Experiments in a controlled laboratory setting with healthy volunteers yielded R-peak detection accuracy above 97 % and mean RR-interval errors below 5 ms. On a dataset of 90 labeled records (45 normal and 45 abnormal) used for training and 8-fold cross-validation, the *CART* classifier achieved an average accuracy of 84.44 %, an F1-macro of 84.38 %, recall values of 77.78 % for the normal class and 91.11 % for the abnormal class, and an AUC-ROC of 85.83 %. The web interface, developed with Bootstrap and Plotly, enables remote review of ECG traces and the export of PDF reports.

This work shows that reliable preliminary cardiac diagnoses can be delivered with low latency and reduced cost (hardware below 500 USD, hosting below 5 USD/month), making the system suitable for primary care and telemedicine in resource-constrained or limited-connectivity settings. Strengths, limitations, and future research lines are discussed, including multichannel support, hybrid modeling, and advanced motion-artifact correction. The emphasis on interpretable algorithms and a complete pipeline—from acquisition to visualization—seeks to facilitate clinical adoption and foster collaboration between engineering and medicine.

Índice general

Agradecimientos	ii
1 Introducción	1
1.1 Contexto mundial y relevancia clínica	1
1.2 Motivación del proyecto	5
1.3 Planteamiento del problema	7
1.4 Objetivos	9
1.5 Alcances y limitaciones	9
2 Fundamentos de cardiopatías y sistemas de asistencia para su diagnóstico	13
2.1 Anatomía y fisiología cardíaca	13
2.2 Patologías cardiovasculares más frecuentes	14
2.3 ECG de superficie: derivaciones e intervalos clínicos	15
2.4 Sistemas de asistencia al diagnóstico basados en aprendizaje automático	16
2.5 Calidad de señal y consideraciones prácticas	17
2.6 Relevancia para el sistema desarrollado	17
2.7 Principios del electrocardiograma de superficie	18
2.8 Sistemas de asistencia al diagnóstico ECG	19
2.9 Clasificación de cardiopatías	24
2.10 Sistemas de diagnóstico de cardiopatías	27
3 Procesamiento de señales ECG y extracción de características	32
3.1 Adquisición y configuración del hardware	32
3.2 Parámetros de muestreo y criterios de calidad	36
3.3 filtrado digital	37
3.4 Preprocesamiento y filtrado digital	37
3.5 Detección de ondas y puntos característicos	39
3.6 Extracción de características	39
3.7 Control de artefactos y criterios de exclusión	42
3.8 Formato de archivos y persistencia	43
3.9 Protocolo de adquisición de la base de datos	44
3.10 Almacenamiento y etiquetado	47
3.11 Selección del hardware <i>OpenBCI GanGLion</i>	47
4 Aprendizaje automático y árboles de decisión	51
4.1 Fundamentos del aprendizaje automático supervisado	51

4.2	Partición, validación y evitación de fugas de información	53
4.3	Balanceo de clases	54
4.4	Modelos considerados	56
4.5	Selección de hiperparámetros y umbral de decisión	60
4.6	Métricas de desempeño	61
4.7	Explicabilidad e interpretación	62
4.8	Persistencia del modelo y paquete de despliegue	64
4.9	Evaluación del clasificador: descripción del protocolo	65
4.10	Buenas prácticas y reproducibilidad	65
4.11	Resumen del capítulo	66
5	Desarrollo del sistema de diagnóstico cardiovascular asistido	67
5.1	Requerimientos del Sistema	67
5.2	Arquitectura general del sistema	71
5.3	Módulo de captura en tiempo real (Python + BrainFlow)	74
5.4	Componente web (PHP + MySQL + Plotly)	81
5.5	Sesiones de revisión clínica e integración hospitalaria	89
5.6	Preparación del conjunto de datos	90
5.7	Módulo de entrenamiento y predicción (Python)	93
5.8	Rendimiento, escalabilidad y operación	98
5.9	Seguridad, privacidad y cumplimiento	99
5.10	Monitoreo, bitácoras y manejo de errores	100
5.11	Despliegue y operación	101
5.12	Mantenibilidad y trazabilidad	102
5.13	Plan de pruebas	103
5.14	Plantillas de figuras y tablas	104
5.15	Resumen del capítulo	107
6	Evaluación experimental y discusión de resultados	108
6.1	Diseño general del experimento	108
6.2	Conjunto de datos y particiones	111
6.3	Métricas y análisis estadístico	113
6.4	Resultados: calidad de señal y detección	113
6.5	Validación clínica	125
6.6	Limitaciones	125
6.7	Discusión	126
7	Conclusiones y trabajo futuro	135
7.1	Síntesis de aportaciones	135
7.2	Cumplimiento de objetivos	136
7.3	Hallazgos principales	137
7.4	Optimización	137
7.5	Trabajo futuro	138
7.6	Recomendaciones operativas	141
7.7	Cierre	141

Índice de figuras

2.1	Componentes principales del electrocardiograma de superficie: ondas P, Q, R, S y T, así como los segmentos PR y ST y los intervalos PR y QT.	19
2.2	Electrocardiógrafo comercial y su trazado	20
2.3	Derivaciones I, II y III	22
2.4	Registro electrocardiográfico estándar de 12 derivaciones (extremidades, derivaciones aumentadas y precordiales) utilizado como referencia clínica en este trabajo.	24
3.1	Placa <i>OpenBCI Ganglion</i> empleada en el prototipo desarrollado.	33
3.2	Esquema ilustrativo de la colocación de los electrodos RA, LA y LL en el tórax para obtener una derivación equivalente a las derivaciones de las extremidades. En este trabajo se utilizan RA y LL como polos de la derivación II, junto con un electrodo de referencia (RL) conectado a D_G.	35
3.3	Electrodos en miembros inferiores y superiores	46
3.4	Electrodos precordiales V1–V6	46
3.5	Prototipo 0	48
4.1	Diagrama de flujo del algoritmo CART para construir un árbol de decisión.	57
4.2	Diagrama de flujo del entrenamiento y predicción de un bosque aleatorio (Random Forest).	58
4.3	Ejemplo de árbol de decisión para clasificación. Cada ruta raíz–hoja define una regla SI–ENTONCES mediante la concatenación de las condiciones de los nodos.	64
5.1	Diagrama de bloques de las funciones principales del sistema.	69
5.2	Arquitectura de dos capas: adquisición y preprocesamiento local (Python + BrainFlow) y, en el servidor, almacenamiento, visualización web y aprendizaje automático.	72
5.3	Diagrama de flujo de punta a punta del sistema: desde la adquisición de la señal ECG con la placa OpenBCI Ganglion, pasando por el preprocesamiento, extracción de características y persistencia local, hasta el envío al servidor, la visualización web y el entrenamiento y predicción del modelo de aprendizaje automático.	73
5.4	Configuración ECG 3-lead con Ganglion	75
5.5	Ejemplo de puntos e intervalos en ECG	79

5.6	Visualización web de un registro ECG con malla clínica y marcadores para ondas P/QRS/T. El panel lateral muestra las características calculadas y los diagnósticos registrados.	86
5.7	Flujo del módulo de notificación y carga de documentos: (1) el clasificador detecta una condición anormal; (2) el sistema envía un correo con un enlace a la página de carga; (3) el paciente sube un ECG de 12 derivaciones; y (4) el médico revisa el documento y registra su revisión clínica.	87
5.8	Sesión 1 con el equipo clínico	90
5.9	Sesión 2 con el equipo clínico	90
5.10	Flujo del módulo Python de entrenamiento y predicción	93
5.11	Árbol de decisión representativo extraído del bosque aleatorio utilizado en producción (profundidad limitada a tres niveles). Cada nodo muestra la característica de corte, el índice de Gini, la proporción de muestras y la clase predominante (Normal o Anormal).	97
5.12	Interfaz de captura en Python (PyQtGraph) con contador y rejilla clínica. En la parte central se observa la traza de ECG filtrada en tiempo real, dibujada sobre una cuadrícula similar a la del papel de un electrocardiógrafo. En la parte superior se muestra un contador de tiempo, y en la inferior se despliegan mensajes de estado (calidad de señal, conexión, etc.), lo que permite al operador verificar, de un vistazo, que la captura se está realizando de forma correcta.	105
5.13	Gráfica ECG interactiva (Plotly) con anotación de ondas. La señal se presenta sobre una malla clínica, y se superponen marcadores que indican la posición aproximada de las ondas P, los complejos QRS y las ondas T. La vista es interactiva: el médico puede hacer <i>zoom</i> , desplazarse por la traza y encender o apagar capas de información, lo cual facilita la revisión detallada de segmentos concretos del ECG.	106
6.1	Matriz de confusión OOF para el clasificador Normal/Anormal utilizando el umbral óptimo $t^* \approx 0,43$. Se observan 35 estudios normales correctamente clasificados, 10 normales etiquetados como anormales, 4 anormales clasificados como normales y 41 anormales correctamente detectados.	115
6.2	Curva ROC en validación cruzada OOF. El área bajo la curva (AUC-ROC) es de 0.858 (85.83 %), lo que indica buena capacidad discriminante para separar estudios normales de anormales al variar el umbral de decisión.	116
6.3	Importancia relativa de características utilizada por el modelo de IA (normalizada). Este gráfico apoya la auditoría clínica al identificar qué variables contribuyen más a la separación Normal/Anormal.	117
6.4	Corrida preliminar (conjunto balanceado). Panel de la plataforma web mostrando métricas con entrenamiento de 84 registros totales (42 normales y 42 anormales). Esta corrida se utilizó como referencia para depuración de parámetros y validación del flujo.	119

6.5	Corrida preliminar (conjunto desbalanceado). Panel de la plataforma web mostrando métricas con entrenamiento de 129 registros totales (42 normales y 87 anormales). Esta prueba evidenció cambios en las métricas al variar la distribución de clases y motivó el uso de estrategias de balanceo y validación cruzada.	120
6.6	Matriz de confusión (CV OOF) del modelo final. Se observan 35 normales correctamente clasificados, 10 normales etiquetados como anormales, 4 anormales clasificados como normales y 41 anormales correctamente detectados. .	121
6.7	Curva ROC (CV OOF) del modelo final. Se obtiene $AUC = 0,858$, indicando buena capacidad discriminante para separar estudios normales de anormales al variar el umbral.	122
6.8	Curva Precisión–Recall (CV OOF) del modelo final. Se obtiene $AUC-PR = 0,820$ y $AP = 0,823$, con línea base 0.50 (conjunto balanceado).	123
6.9	Árbol representativo extraído del bosque aleatorio (profundidad 3). Cada nodo muestra característica de corte, índice de Gini, proporción de muestras y clase predominante.	124

Índice de tablas

1.1	Diez principales causas de defunción en México, enero–marzo 2024	3
1.2	Cinco principales causas de defunción en San Luis Potosí, enero–marzo 2024	3
2.1	Trazabilidad entre características temporales del ECG y su utilidad clínica. .	16
2.2	Comparación de categorías portátiles de ECG	30
3.1	Comparación básica de alternativas exploradas.	48
5.1	Esquema resumido de <code>registros_ecg</code> (campos clave)	85
5.2	Contrato de <code>actualizar_intervalos.php</code> (POST <code>x-www-form-urlencoded</code>). Cada fila muestra el nombre de la clave esperada por la API, el tipo de dato y un ejemplo numérico o textual basado en valores típicos de un adulto en reposo.	106
6.1	Resumen del conjunto de datos utilizado en este estudio.	111
6.2	Umbral óptimo y matriz de confusión (OOF) para el modelo final Random Forest.	118
6.3	Rúbrica de comparación (criterios cualitativos y cuantitativos) entre el prototipo propuesto, Apple Watch (ECG) y el ECG clínico de 12 derivaciones. . .	131

Capítulo 1

Introducción

1.1. Contexto mundial y relevancia clínica

Las enfermedades cardiovasculares (ECV) constituyen, desde hace más de dos décadas, la primera causa de mortalidad y pérdida de años de vida saludable a escala planetaria. Según la Organización Mundial de la Salud, en 2022 se registraron aproximadamente 17.9 millones de defunciones atribuibles a estas patologías, cifra que representa cerca del 32 % de todas las muertes globales [1]. Aún más preocupante resulta el hecho de que más del 75 % de estos decesos ocurrieron en países de ingresos bajos y medios, donde los sistemas de salud carecen de los recursos necesarios para llevar a cabo tamizajes tempranos y ofrecer tratamientos oportunos [2].

Las proyecciones del Institute for Health Metrics and Evaluation advierten que, de no implementarse intervenciones costo-efectivas a gran escala, la mortalidad cardiovascular absoluta podría incrementarse en un 22 % para el año 2030 [3]. Este panorama obliga a reorganizar las estrategias de prevención y diagnóstico, incorporando tecnologías de bajo costo y alto impacto poblacional.

En el contexto nacional, México no escapa a esta tendencia. El boletín más reciente del Instituto Nacional de Estadística y Geografía (INEGI) reporta que las ECV encabezaron la estadística de mortalidad en 2023 con 100 710 defunciones, equivalentes al 21.3 % del total de muertes registradas [4]. La Secretaría de Salud estima que el costo anual de atención y rehabilitación de pacientes con cardiopatías supera los 80 000 millones de pesos, lo que equivale aproximadamente al 0.3 % del PIB nacional [5]. Cuando se incluyen los años de productividad perdidos, el impacto económico y social prácticamente se duplica [6].

El estado de San Luis Potosí presenta indicadores aún más críticos: la tasa bruta de mortalidad por ECV alcanzó 334 defunciones por cada 100 000 habitantes en 2023, mientras que la tasa estandarizada se situó en 315/100 000, ambas por encima de la mediana nacional (322 y 307, respectivamente) [4]. Estos valores reflejan una transición demográfica acelerada y una alta prevalencia de factores de riesgo modificables, tales como hipertensión arterial, dislipidemia, tabaquismo y sedentarismo [7].

Frente a esta realidad, resulta imperativo contar con herramientas diagnósticas que sean accesibles, portátiles y de bajo costo, capaces de detectar alteraciones en la actividad eléctrica cardíaca *in situ*. La detección temprana de anomalías en el electrocardiograma puede reducir significativamente los tiempos de referencia a atención especializada y, por ende, mejorar el

pronóstico de los pacientes en entornos de atención primaria.

Epidemiología de las enfermedades cardiovasculares

La epidemiología moderna de las ECV se caracteriza por cuatro rasgos clave:

1. **Transición demográfica acelerada.** El envejecimiento poblacional, fruto de la mejora en la esperanza de vida y la disminución de las tasas de mortalidad infantil, ha modificado drásticamente la pirámide etaria global. Este fenómeno incrementa de manera exponencial la prevalencia de enfermedades crónicas, entre las que destacan la cardiopatía isquémica y el accidente cerebrovascular, especialmente en adultos mayores de 60 años [8].
2. **Persistencia de factores conductuales.** A pesar de un mayor conocimiento sobre estilos de vida saludables, persiste un alto consumo de alimentos ultraprocesados, ricos en grasas saturadas y sodio, así como niveles alarmantes de sedentarismo. El tabaquismo y el abuso de alcohol continúan siendo vectores de daño vascular: se estima que estas conductas explican hasta el 80 % del riesgo atribuible a ECV en población adulta [9].
3. **Interacción con enfermedades emergentes.** La pandemia de COVID-19 puso de manifiesto la fragilidad del sistema cardiovascular ante infecciones sistémicas. Las personas con ECV previas presentaron tasas de complicación y mortalidad significativamente superiores, revelando una interacción bidireccional: la infección agrava la función cardíaca y, a su vez, la enfermedad subyacente intensifica la respuesta inflamatoria [10].
4. **Carga económica desproporcionada.** En América Latina, el costo directo de la atención de ECV —hospitalizaciones, intervenciones coronarias y rehabilitación— supera los 50 mil millones de dólares anuales. Por cada dólar invertido en prevención primaria (control de hipertensión, programas de cesación tabáquica y promoción de actividad física), se recuperan hasta 3.2 dólares en ahorro de costos sanitarios a mediano plazo [11].

Además de estos factores, la desigualdad en el acceso a servicios de salud agrava la brecha entre regiones urbanas y rurales. En zonas remotas, la falta de equipos diagnósticos básicos impide la detección oportuna de arritmias y isquemia silente, lo que conlleva derivaciones tardías y peores desenlaces clínicos. Por ello, la adopción de herramientas de bajo costo y fácil implementación, como dispositivos portátiles de ECG, aparece como una estrategia fundamental en la reducción de la morbilidad y mortalidad cardiovascular.

El electrocardiograma (ECG) de superficie se elige hoy como la prueba de rastreo más accesible y con evidencia sólida de costo–efectividad. Su capacidad para detectar arritmias prevalentes (fibrilación auricular, flutter), bloqueos de rama y cambios isquémicos agudos lo convierte en un componente esencial de los programas de salud pública destinados a la detección temprana, el monitoreo comunitario y el seguimiento longitudinal de pacientes con alto riesgo cardiovascular [12].

A continuación se presentan dos tablas con las principales causas de muerte en 2024, por entidad:

Tabla 1.1: Diez principales causas de defunción en México, enero–marzo 2024

Ranking	Causa de muerte	Número
1	Enfermedades del corazón	51 586
2	Diabetes mellitus	30 501
3	Tumores malignos	23 464
4	Influenza y neumonía	11 098
5	Enfermedades del hígado	10 292
6	Accidentes	9 480
7	Enfermedades cerebrovasculares	9 237
8	Agresiones (homicidios)	7 137
9	Enfermedades pulmonares obstructivas crónicas	5 819
10	Insuficiencia renal	4 563

Fuente: INEGI, Estadísticas de Defunciones Registradas (EDR), enero–marzo 2024 [4].

Tabla 1.2: Cinco principales causas de defunción en San Luis Potosí, enero–marzo 2024

Ranking	Causa de muerte	Número
1	Enfermedades del corazón	1 425
2	Diabetes mellitus	759
3	Tumores malignos	538
4	Accidentes	254
5	Influenza y neumonía	246

Fuente: INEGI, Estadísticas de Defunciones Registradas (EDR), enero–marzo 2024 [4].

Breve historia del electrocardiograma

El electrocardiograma (ECG) es hoy la herramienta diagnóstica de primera línea en cardiología, pero sus orígenes se remontan a los primeros experimentos de finales del siglo XIX. A continuación se detallan los hitos más relevantes de su evolución:

1. Pioneros y primeras mediciones (1887–1905) El fisiología inglés Augustus Waller publicó en 1887 el primer registro de la actividad eléctrica cardíaca usando un *capillary electrometer*, aunque su trazo era rudimentario y de baja fidelidad. Dieciséis años después, el fisiólogo holandés Willem Einthoven diseñó el *string galvanometer*, un delicado filamento de cobre suspendido en un campo magnético que, al vibrar con las corrientes cardíacas, permitía obtener trazados de alta resolución. En 1903 publicó la forma de la onda en humanos y, tres años más tarde, estableció la nomenclatura P, Q, R, S y T para identificar cada deflexión del ciclo cardíaco [13].

2. De la analogía al estándar clínico (1920–1960) Desde la década de 1920 los aparatos de Einthoven se simplificaron, reemplazando su voluminosa electrónica por sistemas de relojería mecánica y válvulas de vacío. En la posguerra, la aparición de transistores facilitó la construcción de equipos más pequeños y fiables. Durante estos años se definieron las 12 derivaciones estándar —seis en el plano frontal y seis precordiales— que capturan la actividad eléctrica del corazón desde diferentes ángulos, convirtiendo al ECG en el “patrón oro” para el diagnóstico de isquemia, infarto agudo de miocardio y trastornos de conducción.

3. La era digital y los primeros algoritmos (1970–2000) La electrificación y digitalización de la señal ECG en los años setenta permitió su almacenamiento en ordenadores personales y el desarrollo de software de análisis. En 1978 el *Minnesota Code* introdujo un sistema estandarizado de codificación automática de hallazgos ECG en grandes estudios epidemiológicos [14]. Paralelamente, las mejoras en la conversión analógica–digital y el procesamiento por microprocesadores redujeron el ruido y permitieron aplicar filtros en tiempo real, estableciendo las bases de los procesadores de señal modernos.

4. Portátiles y telemedicina (2000–2020) A inicios del siglo XXI surgieron los primeros ECG de bolsillo y monitores holter digitales, capaces de grabar 24–48 h de trazados continuos. La integración de protocolos inalámbricos (Wi-Fi, Bluetooth) y la proliferación de smartphones llevaron el ECG a entornos no clínicos, facilitando la monitorización remota y la telesalud. Dispositivos como AliveCor Kardia o Apple Watch integraron sensores ECG para detección de fibrilación auricular, validando la factibilidad de arritmia-alarma en tiempo real.

5. Sensórica abierta y aprendizaje automático (2020–presente) La democratización de plataformas de hardware abierto, como Arduino o OpenBCI Ganglion, combinada con librerías de procesamiento en Python (BrainFlow, BioSPPy) ha permitido construir sistemas de adquisición personalizados. A la par, los algoritmos de machine learning interpretables—árboles de decisión, bosques aleatorios y, más recientemente, redes neuronales ligeras—se han aplicado al ECG para clasificar patrones de ritmo, cuantificar variabilidad y alertar sobre eventos

críticos en milisegundos. Esta sinergia de hardware accesible y software de libre uso hace posible prototipos de diagnóstico cardiovascular asistido como el que presenta esta tesis.

Visión de futuro Hoy, el ECG continúa evolucionando: la incorporación de inteligencia artificial explicable, algoritmos de aprendizaje federado y análisis multibanda promete una medicina personalizada y predictiva. En este contexto, desarrollar sistemas portátiles, robustos y de bajo costo no es solo un avance tecnológico, sino un paso hacia la equidad en salud, permitiendo llevar diagnósticos de calidad a poblaciones rurales y de bajos recursos.

1.2. Motivación del proyecto

La creciente demanda de diagnóstico y monitoreo cardíaco temprano contrasta con el elevado costo y la complejidad de los equipos convencionales. En las unidades de cuidados intensivos (UCI) y los servicios especializados de cardiología, los monitores multiparámetro de doce derivaciones —que incluyen análisis avanzados de arritmias, isquemia y variabilidad del ritmo— tienen un precio de adquisición que oscila entre 10 000 y 30 000 USD por unidad, sin contar los gastos periódicos de mantenimiento, actualización de algoritmos propietarios y formación continua del personal técnico y médico [15].

Para hospitales de segundo nivel, centros de salud rurales y clínicas comunitarias en México, este coste resulta prohibitivo. Muchas comunidades alejadas de las capitales carecen de especialistas en cardiología y de infraestructura eléctrica estable, lo que obliga al personal de salud a depender de equipos de sobremesa compartidos o métodos manuales de registro ECG, con el consiguiente retraso en la detección de patologías críticas. La precariedad económica y la escasez de médicos capacitados agravan esta brecha, incrementando la mortalidad prevenible por enfermedades cardiovasculares en zonas marginadas.

Además, gran parte del software clínico incorporado en estos sistemas de monitoreo utiliza algoritmos “caja negra” basados en redes neuronales profundas patentadas, lo cual impide al clínico acceder a las reglas de decisión o ajustar umbrales según el perfil epidemiológico local. Esta falta de transparencia complica las auditorías, la adaptación de protocolos y la confianza del usuario en las alertas generadas [16].

Frente a este panorama, la plataforma *OpenBCI Ganglion* emerge como una alternativa abierta y económica (menos de 300 USD). Con cuatro canales analógicos de 24 bits, transmisión Bluetooth Low Energy y un ecosistema de software libre (BrainFlow, PyQt, scikit-learn), permite capturar una derivación estándar II con relación señal-ruido suficiente para identificar complejos QRS, medir intervalos RR y extraer características clínicas de interés.

La filosofía *open source* facilita la auditoría y personalización de cada etapa del procesamiento —desde el filtrado digital hasta la lógica de clasificación—, garantizando transparencia y adaptabilidad a las condiciones locales. En contraste con la solución móvil basada en AD8232 de Sánchez Ovalle (2020), este proyecto desarrolla un “pipeline” completo que corre en hardware de propósito general (MacBook Air y PC con Windows), transfiere de forma segura los resultados a un servidor compartido y presenta al médico una interfaz web interactiva para validación de hallazgos.

El objetivo final es dotar a las comunidades rurales y periferias urbanas de México de una herramienta portátil, de bajo costo y fácil operación, capaz de salvar vidas mediante la

detección temprana de anomalías cardíacas allí donde no hay un especialista a mano y los recursos son escasos.

Limitaciones de los sistemas de diagnóstico actuales

Aunque los sistemas electrocardiográficos de alta gama han demostrado su eficacia en entornos hospitalarios, presentan varias barreras que limitan su adopción y eficiencia en entornos de atención primaria y zonas rurales:

- **Costo y logística:** Los monitores multiparámetro de doce derivaciones suelen requerir una inversión inicial de decenas de miles de dólares, así como contratos anuales de mantenimiento y actualizaciones de software bajo licencias propietarias [15]. Además, estos equipos demandan infraestructura hospitalaria fija, con tomas eléctricas dedicadas y sistemas de respaldo de energía (UPS), lo cual dificulta su despliegue en clínicas de primer nivel o puestos de salud móvil que carecen de suministros eléctricos estables.
- **Portabilidad limitada:** Los ecógrafos y monitores de ECG de 12 derivaciones suelen pesar entre 2 y 4 kg, alojarse en carros rodantes y depender de múltiples cables y electrodos estándar. Su configuración y calibración requieren personal entrenado, lo que retrasa el tiempo de uso en escenarios de urgencia o atención comunitaria. Por otro lado, los dispositivos de bolsillo, aunque más ligeros, suelen registrar sólo segmentos breves (30 s) de un solo canal y ofrecen análisis automático muy básico, sin clasificación de arritmias complejas [16].
- **Falta de interpretabilidad clínica:** La integración creciente de algoritmos de aprendizaje profundo en el post-procesado de señales ECG ha mejorado las tasas de detección de arritmias, pero lo hace a costa de la transparencia. Estos “modelos de caja negra” proporcionan un resultado binario (normal/anormal) sin indicar qué característica de la señal (intervalo QT prolongado, morfología anómala de onda T, variabilidad del ritmo) motivó la alerta. Esto suele generar desconfianza en el clínico y dificulta la validación o ajuste de parámetros según protocolos locales [12].
- **Dependencia de conectividad y seguridad de datos:** Soluciones basadas en la transmisión continua a la nube prometen monitoreo remoto y análisis colaborativo, pero requieren ancho de banda permanente y redes seguras. En zonas rurales con conectividad intermitente, esta dependencia puede resultar inviable. Además, el envío de datos sensibles de pacientes sin cifrado de extremo a extremo incrementa el riesgo de exposición y vulneración de la privacidad, especialmente si los centros de salud no cuentan con políticas y tecnologías robustas de protección de la información [17].

Necesidad de soluciones portátiles e inteligentes

En México, más de 57.5 millones de personas habitan en localidades con menos de 50,000 habitantes, donde la oferta de especialistas en cardiología es insuficiente y las distancias geográficas dificultan el acceso oportuno a servicios de diagnóstico avanzado [5]. A este reto

se suman limitaciones de transporte, escasa infraestructura eléctrica y presupuestos reducidos en los centros de salud rurales.

En este contexto, las herramientas portátiles de bajo coste tienen el potencial de transformar el abordaje de las enfermedades cardiovasculares en tres frentes clave:

1. **Telemedicina síncrona y colaborativa:** Dispositivos ligeros capaces de captar la señal ECG y transmitirla de forma segura —mediante cifrado de extremo a extremo— a cardiólogos en hospitales de referencia. Esta segunda opinión en tiempo real no solo incrementa la precisión diagnóstica, sino que también optimiza el uso de los recursos humanos y evita traslados costosos para el paciente.
2. **Tamizaje comunitario proactivo:** Un sistema portátil puede incorporarse a campañas de salud pública o brigadas médicas itinerantes, permitiendo la detección rápida de bradicardias, taquicardias y arritmias en menos de cinco minutos por paciente. Al identificar anomalías en el lugar de residencia, se facilita la derivación temprana de casos de alto riesgo y se reduce la carga de trabajo en los hospitales, mejorando la distribución de los servicios sanitarios.
3. **Seguimiento domiciliario y empoderamiento del paciente:** Para pacientes con hipertensión, cardiopatías previas o factores de riesgo múltiples, la capacidad de realizar registros ECG diarios en su entorno habitual representa un salto cualitativo en el autocontrol de la enfermedad. Los resultados instantáneos y la retroalimentación automatizada evitan visitas innecesarias y permiten ajustar tratamientos de manera dinámica, favoreciendo la adherencia terapéutica y la educación del paciente.

Con base en estas necesidades, la presente investigación propone un esquema *edge-server* que combina la placa *OpenBCI Ganglion* para la captura de ECG, un conjunto de algoritmos de procesamiento digital de señales (filtrado cero-fase, detección de complejos QRS e identificación de ondas P/T) y un clasificador basado en árboles de decisión. Este enfoque garantiza un diagnóstico preliminar “explicable”, en el que cada predicción se fundamenta en umbrales e intervalos clínicos rastreables, facilitando la validación y confianza por parte de los profesionales de la salud rural.

1.3. Planteamiento del problema

La atención oportuna de las enfermedades cardiovasculares requiere, en primer lugar, un registro electrocardiográfico (ECG) con calidad clínica —es decir, señal con bajo nivel de ruido y resolución suficiente para identificar con fiabilidad los complejos QRS, las ondas P y T y los intervalos temporales relevantes—, y en segundo lugar, su interpretación casi inmediata por parte de personal médico entrenado. En la práctica hospitalaria de segundo y tercer nivel, estos dos requerimientos se satisfacen mediante monitores multiparámetro de doce derivaciones, que incorporan algoritmos avanzados de detección y ofrecen alertas en tiempo real.

Sin embargo, en unidades de atención primaria, clínicas rurales y centros de salud comunitarios, coexisten tres barreras fundamentales:

- **Tecnología insuficiente:** Los dispositivos portátiles que se emplean en estas áreas suelen operar con un único canal de ECG y utilizan amplificadores de baja resolución (8–12 bits), lo cual dificulta la identificación robusta de eventos críticos como arritmias o isquemia transitoria.
- **Falta de análisis automático fiable:** La mayoría de los equipos de bolsillo carecen de software de interpretación integrado o confían en algoritmos de caja negra, con tasas de falsos positivos/negativos que desincentivan su uso clínico.
- **Dependencia de derivación al hospital:** Cuando un paciente presenta síntomas sugestivos de cardiopatía (dolor torácico, disnea, palpitaciones), a menudo debe desplazarse decenas de kilómetros hasta un centro regional, lo que retrasa la toma de decisiones y aumenta el riesgo de complicaciones graves, incluyendo infarto agudo de miocardio o muerte súbita.

En este contexto, surgen las siguientes preguntas de investigación:

1. ¿Cómo se pueden integrar el módulo de adquisición con OpenBCI Ganglion y el procesamiento de señales ECG en un sistema de apoyo al diagnóstico de enfermedades cardiovasculares basado en árboles de decisión, manteniendo la interpretabilidad y la eficiencia computacional?
2. ¿Qué grado de confianza y aceptación muestran los cardiólogos al comparar los diagnósticos preliminares generados por el sistema (NORMAL/ANORMAL) con su propio criterio clínico y la revisión directa de las trazas ECG?

La respuesta a este interrogante podría traducirse en:

- La dotación de equipos capaces de generar una alerta precoz de posibles arritmias o alteraciones de conducción, reduciendo los traslados innecesarios.
- Un respaldo técnico al médico general, mediante características cuantitativas (frecuencia cardíaca, QTc, duración de QRS) expuestas de forma clara y rastreada, que favorecen la toma de decisiones basadas en evidencia.
- Un modelo replicable y escalable que aproveche tecnologías abiertas, minimice los costes de adquisición y mantenimiento, y promueva la autonomía de las comunidades de salud rurales.

De este modo, la investigación pretende cerrar la brecha entre la capacidad diagnóstica de los grandes hospitales y la realidad de las localidades remotas, contribuyendo a disminuir la mortalidad y la discapacidad asociadas a las enfermedades cardiovasculares en México.

1.4. Objetivos

Objetivo general

Desarrollar, implementar y evaluar un sistema de asistencia al diagnóstico de enfermedades cardiovasculares basado en la placa OpenBCI Ganglion y una computadora personal, que integre adquisición en tiempo real, procesamiento digital de la señal ECG y un clasificador interpretable de tipo árbol de decisión (CART), entrenado y calibrado para emitir un diagnóstico preliminar *NORMAL* o *ANORMAL*, con el propósito de apoyar la toma de decisiones clínicas y funcionar como herramienta de alerta temprana en un entorno controlado.

Objetivos específicos

1. **Diseñar e implementar** un módulo de captura de señales ECG con OpenBCI Ganglion, capaz de adquirir y filtrar la señal en tiempo real (filtros pasa-banda y *notch*) en plataformas macOS y Windows.
2. **Extraer** un conjunto de características temporales (intervalos RR, QTc, RMSSD) y morfológicas (duraciones PR, QRS, ST; relaciones de amplitud R/T y R/S) que reflejen el estado eléctrico del corazón y puedan ser utilizadas tanto por el árbol de decisión como por el cardiólogo.
3. **Entrenar y optimizar** un clasificador interpretable de tipo árbol de decisión (CART) usando un conjunto equilibrado de 90 registros etiquetados (45 normales y 45 anormales), aplicando validación cruzada estratificada de ocho pliegues para estimar su desempeño (exactitud, precisión y *recall* por clase, F_1 macro, AUC-ROC, AUC-PR y *average precision*) y seleccionar el umbral de decisión que maximiza el F_1 macro.
4. **Desarrollar** una interfaz web ligera, basada en PHP/MySQL y Plotly, que permita visualizar la señal ECG filtrada, superponer las marcas de ondas y mostrar al médico indicadores clave (BPM, QTc, diagnóstico preliminar) de forma clara, comprensible y rastreadable.
5. **Evaluar y validar** el desempeño del sistema en un entorno controlado, midiendo características de eficiencia (tiempo de captura, latencia de inferencia) y recabando la opinión y el grado de confianza de cardiólogos y médicos generales mediante encuestas y entrevistas estructuradas.

1.5. Alcances y limitaciones

El presente estudio se focaliza en señales obtenidas mediante una única derivación estándar II, capturadas en sujetos en reposo durante intervalos de 30 segundos. Este acotamiento responde a la necesidad de garantizar condiciones uniformes de medición y se ajusta a las capacidades de adquisición del dispositivo OpenBCI Ganglion en entornos de atención primaria.

El sistema no pretende reemplazar la interpretación clínica de cardiólogos o personal médico capacitado, sino ofrecer un *filtro previo* y una alerta temprana para priorizar la atención de pacientes con posibles anomalías cardíacas (bradicardia, taquicardia, arritmias). El diagnóstico generado por el árbol de decisión debe entenderse como una orientación preliminar que siempre se complementa con la evaluación clínica confirmatoria.

La validación inicial se diseñó como un estudio piloto con 30 pacientes ambulatorios en condiciones de consulta rutinaria. Estos resultados aportan evidencias sobre la factibilidad técnica y la aceptabilidad clínica del prototipo. No obstante, la generalización a poblaciones más heterogéneas —incluyendo pacientes con movimiento involuntario, esfuerzo físico o posturas distintas— requerirá ampliar la muestra y emplear múltiples derivaciones (I, II, III y precordiales), aspectos que se proponen como trabajo futuro.

En consecuencia, esta sección delimita únicamente el alcance actual del sistema. Las principales limitaciones metodológicas —variabilidad intersujeto, ruido ambiental, dependencias de conectividad y retos de escalabilidad e integración con expedientes electrónicos— se describen con mayor detalle en el capítulo de *Trabajo futuro*, junto con las acciones necesarias para superarlas.

Contribuciones de esta tesis

Esta tesis presenta aportes técnicos y clínicos orientados al desarrollo e integración de un prototipo de apoyo al diagnóstico basado en ECG, con enfoque interpretable y validación interna en un entorno controlado:

1. **Pipeline de procesamiento y caracterización ECG (robusto e interpretable).** Se implementa un algoritmo de **detección de picos R robusto** frente a ruido de red (60 Hz) y artefactos de movimiento, basado en una variante de los métodos de Pan–Tompkins/Hamilton–Tompkins con ventanas adaptativas y una validación básica de calidad de señal. A partir de latidos confiablemente detectados, el sistema calcula de forma consistente un **vector de características clínicas alineado con la práctica cardiológica**, que incluye intervalos RR, PR, QRS, QT, QTc y ST, así como frecuencia cardíaca (BPM) y medidas de variabilidad (p. ej., RMSSD). Este diseño reduce errores por detección espuria y facilita que las variables sean interpretables tanto por el clasificador como por el cardiólogo. Adicionalmente, se documenta la viabilidad de una solución de bajo costo (OpenBCI Ganglion + computadora convencional), con un costo de hardware menor a 500 USD y hospedaje web menor a 5 USD/mes, disminuyendo la barrera económica frente a equipos comerciales (por encima de 10 000 USD).
2. **Conjunto de datos curado, anotado y revisado clínicamente, con evidencia de aceptación.** Se construye un **conjunto de datos balanceado** de 90 registros de ECG (45 normales y 45 anormales), acompañado de metadatos clínicos relevantes (edad, sexo y antecedentes) y revisado por un cardiólogo. Este dataset, generado con la plataforma propuesta, sirve como base estructurada para entrenamiento y comparación de modelos en trabajos posteriores, bajo resguardo ético y uso exclusivo de investigación y validación interna. Además, se documenta la **concordancia y aceptación clínica** mediante un protocolo de evaluación que incluye revisión sistemática

de registros, comparación con el diagnóstico automático y recolección del grado de confianza y comentarios cualitativos sobre los resultados del sistema.

3. **Clasificador interpretable CART con calibración del umbral y métricas auditable.** Se entrena y calibra un clasificador de tipo **árbol de decisión (CART)** mediante validación cruzada estratificada de ocho pliegues, seleccionando el umbral que maximiza el F_1 macro. Con los 90 registros etiquetados, el modelo alcanza una exactitud media de 84.44 %, un F_1 macro cercano a 0.84, *recall* del 77.78 % para la clase Normal y del 91.11 % para la clase Anormal, así como AUC-ROC y AUC-PR superiores a 0.82. El sistema pone a disposición del clínico curvas ROC/PR, matriz de confusión e importancia de características para facilitar la auditoría de decisiones del modelo.
4. **Arquitectura e interfaz integradas para trazabilidad y visualización clínica.** Se desarrolla una **arquitectura híbrida *edge-server*** que combina procesamiento local (Python/BrainFlow) para filtrado, extracción de características e inferencia, con un backend ligero (PHP+MySQL) en servidor compartido, mejorando la trazabilidad de entradas, parámetros y salidas. Sobre esta arquitectura se implementa una **interfaz web interactiva** basada en Bootstrap y Plotly, que permite explorar la traza ECG con cuadrícula clínica, marcas de ondas e intervalos, y revisar características derivadas (BPM, QTc, diagnóstico preliminar, ritmo y anomalías) desde cualquier navegador. La interfaz integra el diagnóstico del médico y el de la IA, facilitando comparación, revisión clínica y exportación de reportes en PDF.
5. **Evolución respecto a la tesis de Sánchez Ovalle (2020).** Se amplía el alcance de una aproximación previa [18], evolucionando de una aplicación móvil de un solo canal, sin filtrado avanzado ni clasificación automática integrada, hacia un *pipeline* completo con filtrado en tiempo real, extracción sistemática de intervalos, clasificación interpretable y almacenamiento centralizado, aumentando la robustez, la explicabilidad y la escalabilidad de la solución.

Organización del documento

Este documento se estructura en siete capítulos que guían al lector desde los fundamentos teóricos hasta la implementación práctica y la evaluación del sistema propuesto:

- **Capítulo 1 – Introducción:** Contextualiza la relevancia clínica y epidemiológica de las enfermedades cardiovasculares, revisa la historia y motivación del electrocardiograma y presenta una visión general del sistema como *pipeline* de extremo a extremo (captura con OpenBCI Ganglion → filtrado digital → detección de ondas → extracción de características clínicas → árbol CART con umbral calibrado → API web → panel de visualización). En este capítulo se formulan el problema de investigación, las preguntas, los objetivos generales y específicos, y los alcances de la tesis.
- **Capítulo 2 – Fundamentos de cardiopatías y sistemas de asistencia para su diagnóstico:** Revisa en profundidad la fisiología cardíaca, la señal ECG y los intervalos clínicos de interés (RR, PR, QRS, QT/QTc, ST) junto con su valor diagnóstico.

Además, se analizan distintos sistemas de apoyo al diagnóstico basados en aprendizaje automático (SVM, k-NN, árboles de decisión, *random forest*), poniendo énfasis en la necesidad de modelos interpretables en entornos clínicos.

- **Capítulo 3 – Procesamiento de señales ECG y extracción de características:** Describe el flujo de preprocesamiento digital (filtros pasa-banda, *notch*, detrend), los algoritmos de detección de QRS y marcado de ondas inspirados en Hamilton–Tompkins, y el cálculo sistemático de los intervalos RR, PR, QRS, QT/QTc, ST y BPM. También se documenta el uso de la placa OpenBCI Ganglion y las bibliotecas oficiales de BrainFlow para la adquisición en tiempo real.
- **Capítulo 4 – Aprendizaje automático y árboles de decisión:** Presenta la preparación del conjunto de datos (90 registros etiquetados, 45 normales y 45 anormales), las estrategias de partición y balanceo, y el entrenamiento del clasificador CART con validación cruzada estratificada. Se discuten las características empleadas (exactitud, precisión, *recall*, F_1 , AUC-ROC, AUC-PR) y el ajuste del umbral de decisión, situando el desempeño de CART frente a alternativas como *random forest*, SVM y k-NN.
- **Capítulo 5 – Desarrollo del sistema de diagnóstico cardiovascular asistido:** Documenta la arquitectura híbrida *edge-server* en dos capas: procesamiento local con Python/BrainFlow para filtrado, extracción de características e inferencia, y backend web PHP/MySQL/Plotly para almacenamiento, consulta y visualización. Se presentan los diagramas de bloques, los módulos de captura, preprocesado, clasificador y API, así como los mecanismos de seguridad, integridad de datos y registro de auditoría.
- **Capítulo 6 – Evaluación experimental y discusión de resultados:** Detalla el protocolo de pruebas en condiciones controladas y la validación clínica con apoyo de un cardiólogo. Se reportan los resultados cuantitativos por pliegue y agregados (media \pm desviación estándar), las curvas ROC/PR y las matrices de confusión, y se integra el grado de confianza del especialista respecto a los diagnósticos automáticos. Finalmente, se compara el desempeño del sistema con trabajos afines y con la tesis de Sánchez Ovalle.
- **Capítulo 7 – Conclusiones y trabajo futuro:** Resume los principales aportes técnicos y clínicos de la tesis, sintetiza las limitaciones detectadas y plantea líneas de mejora, tales como la extensión a ECG multicanal, el uso de modelos híbridos o en ensamble, el incremento de la robustez frente a artefactos de movimiento y el despliegue ampliado en escenarios de telemedicina y primer nivel de atención.

Capítulo 2

Fundamentos de cardiopatías y sistemas de asistencia para su diagnóstico

2.1. Anatomía y fisiología cardíaca

El corazón humano puede imaginarse como una bomba de cuatro compartimentos: dos aurículas (superiores) y dos ventrículos (inferiores). Su tarea principal es mantener en movimiento la sangre por todo el cuerpo y hacia los pulmones. El ventrículo izquierdo envía sangre oxigenada a la circulación sistémica, es decir, a todos los órganos; el ventrículo derecho la dirige a los pulmones para que se vuelva a oxigenar. En reposo, un adulto sano moviliza alrededor de 5 L min^{-1} de sangre, aunque este valor puede aumentar varias veces durante el ejercicio o ante situaciones de estrés [19].

Para que esta bomba funcione de manera ordenada, el corazón cuenta con un sistema eléctrico propio. El *nodo sinusal*, situado en la aurícula derecha, actúa como marcapasos natural y genera impulsos entre 60 y 100 latidos por minuto. Cada impulso se propaga primero por las aurículas, llega al *nodo auriculoventricular*, desciende por el *haz de His* y se distribuye finalmente a través de las fibras de Purkinje por todo el músculo ventricular. Este recorrido garantiza que las cavidades se contraigan en el orden correcto y con buena coordinación: primero las aurículas llenan a los ventrículos y luego los ventrículos expulsan la sangre hacia la circulación. Cuando este sistema de conducción se altera —por ejemplo, por automatismo anómalo, reentrada o bloqueo— aparecen arritmias y trastornos de conducción que pueden verse en el electrocardiograma (ECG) [20].

Parámetros electrofisiológicos de referencia. En la práctica clínica se utilizan ciertos intervalos como “reglas básicas” para valorar si el ritmo es normal. En adultos sanos, el intervalo PR suele encontrarse entre 120 y 200 ms, la duración del complejo QRS entre 80 y 100 ms y el QT corregido (QTc) suele ser $< 450 \text{ ms}$ en hombres y $< 470 \text{ ms}$ en mujeres. Cuando estos valores se alargan o acortan de forma mantenida, orientan hacia problemas de conducción (por ejemplo, PR prolongado o QRS ancho) o hacia alteraciones de la repolarización (QTc anómalo). Lo importante para esta tesis es que todos estos parámetros pueden medirse incluso con un registro de una sola derivación [21].

Relevancia para este trabajo. El sistema desarrollado en esta investigación registra un ECG de superficie de una sola derivación y, a partir de él, calcula automáticamente varias características temporales: RR, PR, QRS, QT/QTc, ST y la frecuencia cardíaca (BPM). Comprender qué representa cada una ayuda a interpretar los resultados del sistema:

- RR/BPM describen el *ritmo* y qué tan rápido late el corazón.
- PR resume la *sincronía* entre aurículas y ventrículos.
- QRS refleja la *activación ventricular*.
- QT/QTc y ST informan sobre la *repolarización* y posibles signos de isquemia o riesgo arrítmico.

En los capítulos dedicados al procesamiento de señal y al aprendizaje automático, estas mismas variables se organizarán en un vector de características que alimenta al clasificador, de modo que el lector pueda seguir el puente directo entre la fisiología básica y las decisiones del modelo.

2.2. Patologías cardiovasculares más frecuentes

Entre las entidades de mayor prevalencia e impacto clínico destacan:

- **Síndrome coronario agudo (SCA):** engloba angina inestable e infarto agudo de miocardio. Cursa con dolor torácico, biomarcadores elevados y cambios isquémicos en el ECG (elevación o depresión del segmento ST, inversión de la onda T) [22].
- **Arritmias supraventriculares:** fibrilación auricular, aleteo auricular y taquicardias paroxísticas. Se manifiestan como irregularidad del intervalo RR (fibrilación) o ritmos rápidos regulares que suelen apreciarse con claridad en derivación II; con frecuencia requieren registros prolongados o *Holter* [23].
- **Bloqueos de conducción:** bloqueo auriculoventricular (grado I–III) y bloqueos de rama (derecha/izquierda). El alargamiento del intervalo PR sugiere enlentecimiento AV; el ensanchamiento del QRS suele indicar bloqueo de rama o conducción intraventricular anómala [24].
- **Cardiopatía hipertensiva y valvulopatías:** hipertrofia ventricular izquierda con alteraciones de repolarización (ondas T negativas, ST descendido) y estenosis/insuficiencia valvular (aórtica o mitral) con repercusión hemodinámica y cambios eléctricos indirectos [25].

Lectura guiada por el ECG. Estas entidades comparten un componente eléctrico que el ECG de superficie permite detectar de forma temprana. En particular: (i) el patrón del segmento ST y la morfología de la onda T orientan a isquemia aguda; (ii) la regularidad o irregularidad de los intervalos RR y la presencia de ondas P apoyan la clasificación de arritmias; y (iii) los intervalos PR y QRS delimitan trastornos de conducción. De ahí el valor

de sistemas portátiles con análisis automático, trazabilidad de características y visualización para el clínico.

En el presente trabajo, el prototipo desarrollado integra estas mediciones para emitir una clasificación preliminar (*normal/anormal*) explicable a partir de reglas basadas en intervalos, lo que facilita su uso en atención primaria y en escenarios con recursos limitados.

2.3. ECG de superficie: derivaciones e intervalos clínicos

El electrocardiograma (ECG) registra, desde la piel, la proyección de los vectores de despolarización y repolarización cardíaca sobre distintos ejes (*derivaciones*). Cada derivación “mira” al corazón desde un ángulo diferente y, aun trabajando con una sola derivación (p. ej. II), la señal temporal permite estimar intervalos y duraciones con valor diagnóstico [21].

Derivaciones de uso común. Las derivaciones de las extremidades (I, II, III) conforman el triángulo de Einthoven; la derivación II suele ofrecer una onda P y un complejo QRS de alta definición, por lo que se emplea con frecuencia para monitoreo continuo. En sistemas portátiles de un canal, como el prototipo descrito en esta tesis, esta derivación permite medir con fiabilidad las principales características temporales.

Intervalos clínicos básicos. A continuación se resumen los intervalos que se utilizan de forma rutinaria en la práctica clínica y que también constituyen el *vector de características* del sistema propuesto:

- **RR:** tiempo entre picos R consecutivos. Es el inverso de la frecuencia cardíaca (BPM) y base de la variabilidad del ritmo.
- **PR:** desde el inicio de la onda P hasta el inicio del QRS. Refleja la conducción aurículo-ventricular (prolongado → enlentecimiento AV o bloqueo de primer grado).
- **QRS:** duración del complejo de activación ventricular. Si se ensancha, orienta a bloqueo de rama o trastornos de conducción intraventricular.
- **QT y QTc:** desde el inicio del QRS hasta el final de la onda T. El QTc (corregido por frecuencia a partir de RR) resume la repolarización global (prolongado → mayor riesgo de arritmias; acortado → síndromes menos frecuentes).
- **ST:** segmento isoelectrico pos-QRS. Su elevación o depresión sugiere isquemia aguda o cambios secundarios.

Rangos de referencia útiles. En adultos, y con variación por edad, sexo y tono autonómico, se consideran habituales: PR 120–200 ms, QRS 80–100 ms y QTc < 450 ms en hombres / < 470 ms en mujeres. Desviaciones sostenidas respecto a estos rangos motivan la búsqueda de trastornos de conducción o de repolarización. Para la delimitación del final de la onda T se emplea un criterio semiautomático (pendiente/tangente) y se valida visualmente cuando es necesario.

Aplicación en el prototipo. El sistema desarrollado calcula de forma automática RR, PR, QRS, QT/QTc, ST y BPM a partir de una derivación de superficie, y utiliza dichas

Tabla 2.1: Trazabilidad entre características temporales del ECG y su utilidad clínica.

Característica	Significado fisiológico	Utilidad diagnóstica
RR / BPM	Ritmo y cronotropismo	Bradicardia, taquicardia, irregularidad (FA)
PR	Conducción AV	Bloqueo AV de primer grado
QRS	Activación ventricular	Bloqueo de rama, conducción intraventricular
QT / QTc	Repolarización global	Riesgo arrítmico, fármacos que prolongan QT
ST	Lesión o isquemia aguda	SCA con elevación o depresión del ST

variables como entrada de un clasificador interpretable. La elección favorece: (i) *interpretabilidad clínica* —son características estándar en cualquier reporte ECG— y (ii) *compatibilidad con hardware de un canal* sin comprometer el tiempo de respuesta.

2.4. Sistemas de asistencia al diagnóstico basados en aprendizaje automático

El análisis automático del ECG se ha abordado tanto con algoritmos *clásicos* como con modelos más recientes orientados a grandes volúmenes de datos. Entre los primeros destacan los métodos basados en distancia, como k -NN (vecinos más cercanos), y los modelos de margen máximo, como las máquinas de soporte vectorial (SVM), que suelen funcionar bien cuando el problema se describe con pocos predictores temporales y morfológicos. Entre los modelos basados en conjuntos, los árboles de decisión y sus extensiones en forma de bosques aleatorios ofrecen un equilibrio favorable entre desempeño y explicabilidad [?].

En este trabajo se adopta un clasificador basado en árboles de decisión tipo CART, implementado como bosque aleatorio (*Random Forest*) en Python usando `scikit-learn`. La elección se justifica por las siguientes razones:

1. **Interpretabilidad clínica.** Cada árbol se puede leer como un conjunto de reglas del tipo “*si QTc es mayor que cierto valor y el QRS es ancho, entonces la probabilidad de anormalidad aumenta*”. Estas reglas son explícitas y se pueden contrastar directamente con los intervalos descritos en la Sección 2.3, lo que facilita el diálogo con el cardiólogo.
2. **Compatibilidad con pocos predictores.** El modelo trabaja con un conjunto reducido de variables, todas clínicamente significativas: RR/BPM, PR, QRS, QT/QTc y ST (además de algunas relaciones derivadas), que ya se calcularon en el módulo de procesamiento. No es necesario recurrir a cientos de características morfológicas difíciles de interpretar.
3. **Baja latencia.** Una vez entrenado, el bosque aleatorio evalúa cada nuevo registro en milisegundos. Esto permite ejecutar la inferencia directamente en el equipo que captura la señal, sin depender de cómputo en la nube ni de conexiones de alta velocidad.

Para asegurar *robustez* y *trazabilidad*, el modelo se entrena con un conjunto equilibrado de registros, de modo que la clase *Normal* y la clase *Anormal* aportan el mismo número

de ejemplos. El entrenamiento se realiza mediante validación cruzada estratificada ($k = 8$ pliegues), de forma que en cada pliegue se conserva la proporción de normales y anormales y se obtiene una estimación más estable del rendimiento.

La búsqueda de hiperparámetros (profundidad máxima, número de árboles, tamaño mínimo de hoja, etc.) se lleva a cabo con `GridSearchCV`, optimizando el F1 macro, que resume el equilibrio entre sensibilidad y especificidad de ambas clases. A partir de las probabilidades generadas en validación cruzada se elige de forma automática el umbral de decisión que maximiza ese F1 macro. Finalmente, el desempeño que se publica en la plataforma web y en los capítulos de resultados se resume mediante la **exactitud** (*accuracy*) promedio: el porcentaje de registros correctamente clasificados sobre las predicciones de validación cruzada. De forma interna también se calculan precisión, *recall*, F1 y áreas bajo las curvas ROC y precisión–*recall*, pero solo se reportan las características que se utilizan efectivamente en el sistema para mantener la consistencia entre código y documento.

2.5. Calidad de señal y consideraciones prácticas

La utilidad clínica del ECG depende directamente de la calidad del registro. En un sistema portátil de una sola derivación, como el que se desarrolla en este trabajo, se priorizan las siguientes medidas:

- **Contacto electrodo–piel.** Preparación básica de la zona (limpieza, posible rasurado o ligera abrasión si procede) y fijación estable de los electrodos para reducir artefactos de movimiento.
- **Filtrado apropiado.** Se emplean filtros pasa–banda para atenuar la deriva de línea base y el ruido de alta frecuencia, y un filtro *notch* en 60 Hz (o 50 Hz según la red eléctrica) para eliminar interferencias de la corriente alterna. En esta tesis se utilizan filtros de fase lineal/cero–fase, de modo que no se desplacen los picos ni se distorsionen los intervalos temporales.
- **Muestreo y resolución.** Se utiliza una frecuencia de muestreo suficiente para preservar la morfología del QRS y distinguir con claridad las ondas P y T. De este modo se garantiza que las medidas de PR, QRS, QT y ST sean comparables con los rangos clínicos de referencia.
- **Validación visual.** La interfaz gráfica permite al médico revisar la traza filtrada, las marcas de picos R y los puntos usados para medir PR, QRS, QT y ST. Esta revisión rápida aporta un control de calidad adicional y refuerza la confianza diagnóstica.

En el *pipeline* descrito en el Capítulo 3 estas buenas prácticas se combinan con un algoritmo robusto de detección de QRS antes de calcular los intervalos que alimentan al clasificador.

2.6. Relevancia para el sistema desarrollado

Los intervalos descritos en la Sección 2.3 constituyen el *vector de características* del clasificador propuesto. La integración de:

1. la captura con la placa OpenBCI Ganglion,
2. la extracción automática de RR, PR, QRS, QT/QTc, ST y BPM a partir de una sola derivación de superficie, y
3. un modelo basado en árboles (Random Forest/CART) con umbral de decisión optimizado sobre validación cruzada,

proporciona un flujo completo, desde la señal cruda hasta una etiqueta *Normal/Anormal* explicable para el clínico.

La validación clínica se realiza contrastando las salidas del modelo con el criterio del cardiólogo y revisando las trazas correspondientes. Este enfoque, centrado en un número reducido de variables estándar y en un modelo interpretable, busca facilitar la adopción del sistema en entornos de atención primaria y en escenarios con recursos limitados, donde se requieren herramientas sencillas, trazables y de bajo costo para apoyar el diagnóstico inicial.

2.7. Principios del electrocardiograma de superficie

El electrocardiograma de superficie registra la actividad eléctrica cardíaca mediante electrodos colocados en la piel. Cada derivación (I, II, III y precordiales V_1 – V_6) ofrece una “vista” diferente del vector de despolarización/repolarización miocárdica. La derivación II, utilizada en este estudio, mide la diferencia de potencial entre brazo derecho y pierna izquierda, y es especialmente sensible al complejo QRS y a la onda P debido a su alineación con el eje eléctrico medio del corazón [26].

En el trazado ECG se distinguen cinco componentes principales:

1. **Onda P:** Representa la despolarización auricular; su duración normal no excede los 120 ms.
2. **Complejo QRS:** Corresponde a la despolarización ventricular; su ancho típico es de 80–100 ms.
3. **Segmento ST y onda T:** Indican la repolarización ventricular; la elevación o depresión del ST y la forma de la T son clave para diagnosticar isquemia o lesión miocárdica.
4. **Intervalos RR y QT:** El intervalo RR refleja la frecuencia cardíaca, mientras que el QT (y su corrección QTc) aporta información sobre la duración del potencial de acción ventricular.

La Figura 2.1 ilustra de manera esquemática la localización de cada onda, segmento e intervalo descrito. La calidad diagnóstica de un ECG depende de la precisión en la colocación de electrodos, la fidelidad del hardware de adquisición y el muestreo mínimo de 200–300 Hz para conservar la morfología de ondas rápidas como el QRS [21]. Tales requerimientos guían la selección del OpenBCI Ganglion como plataforma de captación en este trabajo.

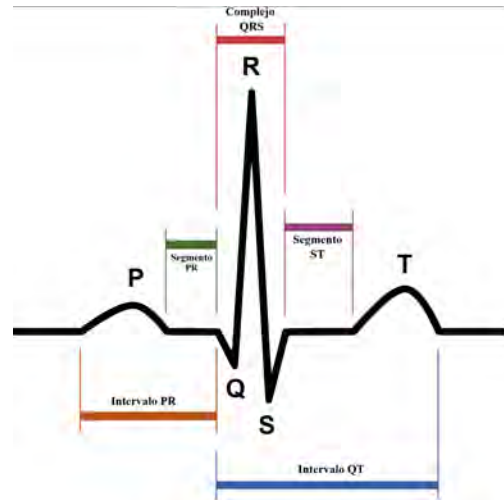


Figura 2.1: Componentes principales del electrocardiograma de superficie: ondas P, Q, R, S y T, así como los segmentos PR y ST y los intervalos PR y QT.

2.8. Sistemas de asistencia al diagnóstico ECG

Durante las últimas décadas se han desarrollado soluciones de apoyo al diagnóstico electrocardiográfico que van desde reglas empíricas codificadas hasta modelos de inteligencia artificial. Las familias más representativas son:

- **Minnesota Code** (años 80). Conjunto de reglas basadas en duraciones y amplitudes de ondas, muy utilizado en estudios epidemiológicos por su estandarización y reproducibilidad [14].
- **Sistemas comerciales de 12 derivaciones**. Integran detectores de arritmias e isquemia con lógica propietaria; para el usuario final suelen funcionar como “caja negra” (no exponen reglas ni umbrales) [27].
- **Wearables y aplicaciones móviles**. Dispositivos de una derivación con algoritmos (en ocasiones de aprendizaje profundo) que entregan salidas binarias *normal/anormal* con escasa explicabilidad clínica [16].
- **Iniciativas abiertas**. Placas como Arduino con AD8232 y librerías de Python que sirven como plataforma educativa; en general carecen de validación clínica amplia y de un análisis en tiempo real robusto [18].



Figura 2.2: Electrocardiógrafo comercial (Welch Allyn) con papel cuadriculado e impresión de derivaciones I, II y III. Se usa como referencia de calidad de señal y de la interfaz tradicional para comparar con el sistema propuesto basado en OpenBCI.

En la Fig. 2.2 se aprecia un equipo portátil estándar que imprime tres derivaciones a la velocidad habitual de 25 mm/s (cada cuadrado pequeño equivale a 0.04 s) y una calibración de amplitud de 10 mm/mV (cuadrado pequeño = 0.1 mV). Este patrón (*gold standard*) sirve para contrastar fidelidad temporal y resolución de amplitud de las señales obtenidas con el sistema propuesto.

El presente proyecto se sitúa en la convergencia de tres ejes: (i) portabilidad del hardware (captura local con OpenBCI), (ii) interpretabilidad algorítmica (árboles CART con reglas legibles) y (iii) validación clínica local. El objetivo es entregar una herramienta transparente, reproducible y asequible para el primer nivel de atención.

Sistema de conducción eléctrica

El sistema de conducción garantiza la generación y la propagación ordenada del estímulo que produce la contracción miocárdica. De forma sucinta:

- **Nodo sinusal (NS).** Marcapasos fisiológico en aurícula derecha; genera impulsos a 60–100 lpm en reposo, modulados por el sistema nervioso autónomo [28].
- **Vías internodales y contracción auricular.** Conducen el impulso hacia el nodo AV mientras las aurículas se despolarizan (onda P) y optimizan el llenado ventricular [?].
- **Nodo auriculoventricular (AV).** Introduce un retardo fisiológico de ~ 100 –120 ms que separa la contracción auricular de la ventricular y protege frente a frecuencias excesivas en caso de arritmias auriculares [29].

- **Haz de His y fibras de Purkinje.** Distribuyen rápidamente el estímulo por ambos ventrículos; el complejo QRS refleja este proceso (típicamente 80–100 ms) [30].

Alteraciones estructurales (infarto, fibrosis), metabólicas (electrolitos) o farmacológicas pueden traducirse en:

- **Bloqueos de rama:** QRS ancho (> 120 ms).
- **Bloqueos AV:** PR prolongado (grado I) hasta disociación AV (grado III).
- **Anomalías del QT:** QTc prolongado o acortado, con implicaciones arrítmicas.

La detección fiable de estas alteraciones requiere registros con colocación estandarizada de electrodos y muestreo ≥ 200 –250 Hz, preservando la morfología de QRS y la línea base [21]. Estos criterios se cumplen en el *pipeline* de captura y preprocesamiento del sistema desarrollado.

Triángulo de Einthoven y derivaciones estándar

El triángulo de Einthoven es el concepto geométrico que sustenta el electrocardiograma de superficie. En 1903, Willem Einthoven demostró que la actividad eléctrica cardíaca puede registrarse colocando electrodos en las extremidades, formando idealmente un triángulo equilátero alrededor del tórax [13]. Cada vértice corresponde a un punto de contacto:

- **Brazo derecho (BD).**
- **Brazo izquierdo (BI).**
- **Pierna izquierda (PI).**

Con estos tres electrodos se definen las derivaciones bipolares originales:

$$\begin{aligned} \text{Derivación I:} & \quad \text{pot}(BI) - \text{pot}(BD) \\ \text{Derivación II:} & \quad \text{pot}(PI) - \text{pot}(BD) \\ \text{Derivación III:} & \quad \text{pot}(PI) - \text{pot}(BI) \end{aligned}$$

Estas tres derivaciones están ligadas por la *ley de Einthoven*:

$$\text{Derivación II} = \text{Derivación I} + \text{Derivación III},$$

relación que ofrece un control de coherencia entre mediciones.

Para ampliar el mapeo del vector cardíaco se incorporaron posteriormente:

- **Derivaciones unipolares aumentadas:** aVR , aVL , aVF , donde el potencial de BD, BI o PI se compara contra un *polo central* virtual formado por el promedio de los otros dos electrodos.
- **Derivaciones precordiales:** $V_1, V_2, V_3, V_4, V_5, V_6$, ubicadas en puntos anatómicos del hemitórax anterior desde el borde esternal derecho (V_1) hasta la línea axilar anterior izquierda (V_6).

Las tres bipolares de miembros, las tres aumentadas y las seis precordiales completan el ECG de 12 derivaciones, considerado el estándar clínico. Cada derivación representa una proyección diferente del vector de despolarización/repolarización, lo que permite:

- Identificar isquemia o infarto comparando cambios de amplitud y morfología en QRS y ST-T entre derivaciones contiguas.
- Detectar bloqueos de rama y alteraciones del eje eléctrico analizando duración, polaridad y dirección de las ondas en distintos planos.
- Diagnosticar arritmias y trastornos de conducción mediante el seguimiento de intervalos (PR, QRS, QT) y patrones repetitivos multicanal.

En la práctica clínica, una correcta colocación de electrodos y una impedancia piel-electrodo baja ($< 5 \text{ k}\Omega$) son esenciales para garantizar calidad diagnóstica. También se recomienda una frecuencia de muestreo mínima de 200 Hz y filtrado en banda de $\approx 0,05 \text{ Hz}$ a 150 Hz (o 0.5 Hz a 40 Hz en escenarios de emergencia), de modo que se preserve la morfología de la señal y se minimicen artefactos y ruido de red [31].

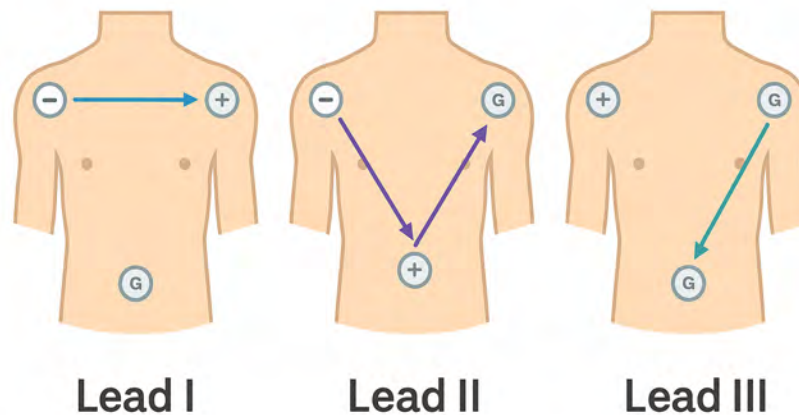


Figura 2.3: Posición de electrodos en extremidades y vectores de las tres derivaciones bipolares estándar de los miembros.

Nota sobre las fuentes. Salvo indicación explícita, los criterios diagnósticos, la nomenclatura de ondas y la disposición de los electrodos para el ECG de 12 derivaciones siguen la tercera edición de [32].

Características de la señal ECG y rango de frecuencias

El trazado electrocardiográfico es la representación temporal de la actividad eléctrica cardíaca y concentra su energía en un espectro relativamente estrecho, coherente con la fisiología del ciclo cardíaco. A efectos prácticos:

- **Frecuencia fundamental (intervalos R–R).** Los tiempos entre picos R consecutivos suelen estar entre 0.6 s y 1.2 s, equivalentes a 0.8–1.7 Hz. Esta banda muy baja captura la variabilidad del ritmo cardíaco (HRV).
- **Componentes de baja frecuencia (ondas P y T).** La despolarización auricular (P) y la repolarización ventricular (T) concentran energía por debajo de 10 Hz. Dado su bajo voltaje y suavidad, un muestreo insuficiente o un filtrado agresivo en bajas frecuencias puede atenuarlas o distorsionarlas.
- **Complejo QRS.** Es el componente de mayor pendiente por la rápida despolarización ventricular. Su espectro útil se extiende típicamente hasta 40 Hz, con armónicos que pueden alcanzar 80–100 Hz en registros de alta resolución. Para preservarlo se emplean filtros paso–banda de 0.5 Hz a 40 Hz o 0.5 Hz a 45 Hz según el escenario.
- **Deriva de línea base (*baseline wander*).** Movimientos respiratorios, cambios posturales y acoples de baja frecuencia introducen componentes por debajo de 0.5 Hz. Se controlan con detrendido y/o filtrado pasa–alto suave (p.ej., $\geq 0,5$ Hz) sin deformar ST y T.
- **Ruido y artefactos.** La interferencia de red (50 Hz o 60 Hz según el país), el electromiograma torácico (> 100 Hz) y los artefactos de movimiento degradan la señal. Se mitigan con filtros *notch* (p.ej., 60 Hz con $Q \approx 30$) y con pasa–bajo para suprimir componentes por encima de 40–45 Hz.

Para cumplir el teorema de Nyquist y conservar la morfología se recomienda muestrear al menos a > 200 Hz, siendo habitual 250 Hz a 500 Hz. Con ello se garantiza que: (i) no haya *aliasing* de componentes hasta 100 Hz, (ii) se preserve la forma del QRS (amplitud y duración) y (iii) se capten con precisión P y T para diagnosticar arritmias auriculares y trastornos de repolarización.

En la práctica, una señal con calidad diagnóstica debe presentar:

- **Relación señal–ruido (SNR)** superior a 10 dB, evitando que el ruido o el EMG oculten la morfología cardíaca.
- **Línea base estable**, con deriva $< 0,5$ mm/s, para identificar con claridad segmentos isoeletrícos (p.ej., ST).
- **Baja interferencia de red**, mediante apantallamiento, toma de tierra adecuada y filtrado *notch* en 50 Hz y 60 Hz.

Estos criterios orientan el diseño del *hardware* (rango dinámico de al menos ± 2 mV, resolución de 12 a 24 bits y *antialiasing* analógico previo al ADC) y del *software* de preprocesamiento, que idealmente aplica filtrado digital en cero fase (p.ej., técnica *filtfilt*) para evitar desplazamientos temporales de los picos.

Tipos de derivaciones y su utilidad diagnóstica

Derivaciones de las extremidades (I, II, III). Proporcionan una vista frontal; la derivación II es la más utilizada para el monitoreo continuo por la claridad con que muestra las ondas P.

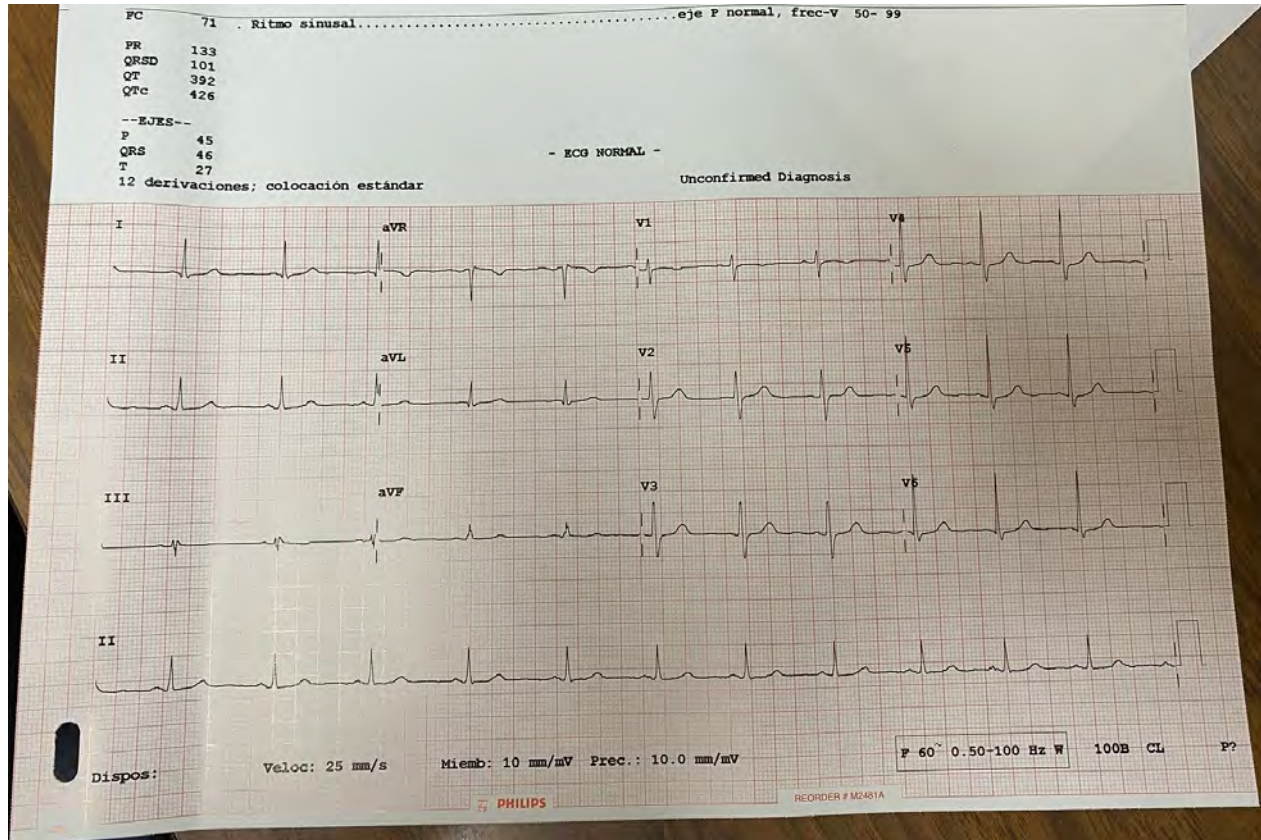


Figura 2.4: Registro electrocardiográfico estándar de 12 derivaciones (extremidades, derivaciones aumentadas y precordiales) utilizado como referencia clínica en este trabajo.

Derivaciones aumentadas (aVR , aVL , aVF). Incrementan la resolución eléctrica en el eje frontal y facilitan la detección de infartos de pared inferior o lateral.

Precordiales (V_1 – V_6). Ofrecen un mapa horizontal del corazón; resultan esenciales para localizar isquemia miocárdica y bloqueos de rama.

Derivaciones modificadas. Configuraciones como la *derivación de Lewis* o *Mason–Lickar* optimizan la detección de arritmias cuando el número de electrodos es limitado, situación frecuente en dispositivos portátiles.

El sistema propuesto en esta tesis trabaja, por simplicidad y consumo energético, con la derivación II —la misma empleada en la mayoría de los monitores de ritmo—, aunque su arquitectura permite escalar a múltiples canales en futuras iteraciones. Como se aprecia en la Figura 2.4, la derivación II forma parte del conjunto completo de 12 derivaciones convencionales.

2.9. Clasificación de cardiopatías

Las enfermedades cardiovasculares abarcan un amplio espectro de trastornos que, a efectos diagnósticos con un electrocardiograma (ECG) de superficie, suelen agruparse en dos grandes

categorías: alteraciones del ritmo cardíaco (arritmias) y patologías del sustrato miocárdico o de la irrigación coronaria (isquemia y lesión de la masa ventricular). A continuación se presentan las entidades más frecuentes dentro de cada grupo, con los hallazgos electrocardiográficos elementales que permiten su detección preliminar.

Arritmias supraventriculares

Las arritmias supraventriculares tienen su origen en estructuras ubicadas por encima del haz de His (aurículas o nodo auriculoventricular). Entre las más relevantes destacan:

- **Fibrilación auricular.** Caracterizada por una actividad eléctrica auricular caótica, sin ondas P identificables y con intervalos RR totalmente irregulares. Es la arritmia sostenida más frecuente en la práctica clínica y se asocia con un elevado riesgo de accidente cerebrovascular isquémico.
- **Flutter auricular.** Producido por un circuito de macro-reentrada en la aurícula derecha, suele manifestarse con ondas F en “diente de sierra” a una frecuencia de aproximadamente 250–350 lpm. La respuesta ventricular depende de la conducción auriculoventricular y habitualmente aparece con relación fija (por ejemplo, 2:1 o 3:1).
- **Taquicardia supraventricular paroxística (TSVP).** De inicio y fin súbitos, originada por reentrada nodal o vía accesoria (síndrome de Wolff-Parkinson-White). Presenta QRS angosto y frecuencia ventricular elevada (>150 lpm), con síntomas de palpitaciones, mareo o síncope.

Arritmias ventriculares

Se originan en el miocardio ventricular y suelen cursar con complejos QRS anchos (> 120 ms) y mayor inestabilidad hemodinámica.

- **Extrasístoles ventriculares (EV).** Latidos prematuros aislados o en salvas, con pausa compensatoria; pueden ser benignos o reflejar irritabilidad miocárdica.
- **Taquicardia ventricular (TV).** Tres o más EV consecutivas a > 100 lpm; QRS ancho y monomorfo o polimorfo. Si hay inestabilidad, requiere cardioversión inmediata.
- **Torsade de pointes.** TV polimorfa asociada a QTc prolongado; QRS “giratorio” en el eje eléctrico. Puede degenerar en fibrilación.
- **Fibrilación ventricular (FV).** Actividad eléctrica desorganizada sin gasto cardíaco efectivo; causa de paro cardíaco súbito. Precisa desfibrilación inmediata.

Bradiarritmias y trastornos de conducción

Incluyen alteraciones del automatismo y de la transmisión del impulso eléctrico.

- **Disfunción del nodo sinusal.** Bradicardia persistente, pausas sinusales o bloqueo sinoauricular; puede producir síncope o fatiga.

- **Bloqueo auriculoventricular (AV).** *Grado I:* PR prolongado (> 200 ms). *Grado II:* conducción intermitente (Mobitz I con alargamiento progresivo del PR; Mobitz II con caídas súbitas del QRS). *Grado III:* disociación AV completa con ritmo de escape.
- **Bloqueos de rama.** QRS ancho (> 120 ms) con morfología característica: derecha (RSRén V_1) o izquierda (complejos anchos en derivaciones laterales).

Isquemia y lesión miocárdica

La reducción del flujo coronario o la necrosis producen cambios de repolarización:

- **Isquemia subendocárdica.** Depresión del ST e inversión de T en derivaciones concordantes con el territorio afectado.
- **Oclusión coronaria aguda.** Elevación persistente del ST (≥ 1 mm en derivaciones contiguas) y, posteriormente, ondas Q patológicas.
- **Cambios dinámicos de la T.** Ondas T “hiperagudas” o pseudonormalización en fases muy precoces del infarto.

Canalopatías y alteraciones de la repolarización

Trastornos eléctricos primarios con ECG típico:

- **QT largo/corto.** QTc prolongado o acortado con riesgo de TV polimorfa o síncope; exige correlación clínica y farmacológica.
- **Síndrome de Brugada.** Elevación del ST en V_1 – V_3 con morfología tipo “coved” y predisposición a arritmias malignas.
- **Preexcitación (WPW).** Intervalo PR corto y onda delta por vía accesoria; puede asociar TSVP o fibrilación auricular preexcitada.

Criterios prácticos utilizados por el sistema

El sistema desarrollado en esta tesis emite un dictamen preliminar (*Normal/Anormal*) apoyándose en características temporales explicables (RR, PR, QRS, QT/QTc, ST y frecuencia). De forma orientativa:

- **Ritmo.** RR < 0.6 s (> 100 lpm) sugiere taquicardia; RR > 1.0 s (< 60 lpm) sugiere bradicardia.
- **Conducción.** PR > 200 ms es compatible con bloqueo AV de primer grado; QRS > 120 ms sugiere bloqueo de rama o conducción intraventricular anómala.
- **Repolarización.** QTc prolongado eleva el riesgo arrítmico; la elevación o depresión del ST orienta a isquemia aguda.

Estos umbrales se emplean como puntos de partida: el clasificador CART ajusta límites y combinaciones en función de los datos disponibles y del equilibrio sensibilidad–especificidad buscado.

Limitaciones del ECG de una derivación

El uso de una sola derivación permite detectar alteraciones globales de ritmo y repolarización, pero no siempre localiza con precisión el territorio isquémico ni caracteriza plenamente la morfología en todo el corazón. Por ello, el sistema se concibe como una herramienta de apoyo y tamizaje: el diagnóstico definitivo queda a cargo del especialista y puede requerir ECG de 12 derivaciones, estudios de laboratorio e imagen complementaria.

2.10. Sistemas de diagnóstico de cardiopatías

En la práctica clínica existen múltiples plataformas para el estudio de la función cardíaca, que van desde equipamiento de alta complejidad instalado en áreas especializadas hasta dispositivos de uso ambulatorio o incluso de consumo masivo. A continuación se describen las principales modalidades.

Electrocardiógrafos de doce derivaciones (12-lead ECG)

Estos equipos constituyen la piedra angular de la evaluación cardiológica. Habitualmente instalados en consultorios y salas de urgencias, registran de forma simultánea:

- **Seis derivaciones del plano frontal** (I, II, III, aVR, aVL, aVF) para evaluar la actividad eléctrica en ejes verticales.
- **Seis derivaciones precordiales** (V_1 - V_6) para analizar fenómenos en el plano horizontal.
- **Cálculo automático** de ejes eléctricos promediados, intervalos (PR, QRS, QT/QTc) y frecuencia cardíaca.

Bajo calibración estándar (velocidad de papel 25 mm/s y amplitud 10 mm/mV), su precisión diagnóstica permite diferenciar síndromes coronarios agudos, bloqueos de rama, hipertrofias auriculares o ventriculares y diversas canalopatías. No obstante, su uso implica:

- **Personal entrenado:** correcta colocación de electrodos, verificación de impedancias y calibración.
- **Infraestructura fija:** suministro eléctrico confiable y espacio acondicionado.
- **Costos elevados:** los equipos clínicos de calidad suelen situarse entre \$10 000 y \$25 000 USD, más mantenimiento y actualizaciones.

En esta tesis, los registros de 12 derivaciones se consideran referencia para contrastar la fidelidad de señal y la interpretación frente al sistema propuesto.

Holter de 24–48 horas

El Holter ambulatorio complementa el ECG puntual con un registro prolongado:

- **Entre 3 y 7 derivaciones** conectadas a un grabador portátil alimentado por baterías.
- **Monitoreo continuo durante 1–2 días**, útil para documentar arritmias intermitentes (p. ej., extrasístoles, fibrilación auricular paroxística) y correlacionarlas con síntomas.
- **Análisis posterior mediante software** que clasifica eventos por tipo y frecuencia, generando reportes de carga arrítmica.

Aunque amplía la ventana diagnóstica, presenta limitaciones: el paciente debe portar el dispositivo todo el tiempo, existe riesgo de artefactos por movimiento y se requiere personal capacitado para la revisión de trazas y la validación del informe.

Monitores multiparámetro en cuidados críticos

En unidades de cuidados intensivos y quirófanos se emplean sistemas integrados que combinan:

- **ECG continuo** con detección automática de arritmias y alarmas en tiempo real.
- **Presión arterial** invasiva y no invasiva para evaluar la hemodinámica minuto a minuto.
- **Oximetría de pulso, capnografía y otros parámetros** respiratorios y metabólicos.

Estos monitores permiten vigilancia estrecha del paciente crítico, pero su complejidad, la necesidad de personal especializado y su costo (hasta \$50 000 USD por cama equipada) los hacen poco viables fuera del entorno hospitalario centralizado. Esta realidad refuerza la motivación por soluciones portátiles y asequibles como la desarrollada en esta tesis.

Dispositivos portátiles y *wearables*

La disponibilidad de electrónica de bajo consumo, sensores más precisos y conectividad ubicua ha permitido que el registro de ECG salga del hospital y llegue al domicilio, al trabajo o al campo. Esta sección ofrece una visión práctica y accesible de las principales familias de dispositivos portátiles, sus casos de uso y sus límites, con el fin de situar el aporte de esta tesis dentro del panorama actual.

¿Por qué interesan? Permiten detección temprana y seguimiento continuo sin trasladar al paciente a un servicio especializado. En cuadros intermitentes (p. ej., palpitations ocasionales) aumentan la probabilidad de “capturar” el evento y facilitan decisiones oportunas. También empoderan al personal de primer contacto al proporcionar señales y características básicas sin requerir equipamiento costoso.

Familias de dispositivos más frecuentes.

- **Electrocardiógrafos de bolsillo (hand-held).** Unos pocos electrodos integrados, registros de 30-s y reportes inmediatos (p. ej., sospecha de fibrilación auricular). Son útiles para tamizaje y para documentar episodios breves.
- **Relojes inteligentes y parches adhesivos.** Los *smartwatches* combinan PPG y ECG de una derivación para emitir alertas de ritmo irregular; los parches permiten monitoreo continuo de 7–14 días con buena adherencia, a costa de análisis diferido y mayor precio por episodio [?, ?].
- **Accesorios para *smartphone*.** Módulos *plug-and-play* que añaden un canal ECG al teléfono. La experiencia depende de la calidad del electrodo y de la conectividad.
- **Placas abiertas para investigación (p. ej., OpenBCI Ganglion).** Dan acceso a la señal cruda (1–8 canales, 24 bits, BLE) y a un ecosistema flexible (BrainFlow, Python). Son idóneas para prototipos con control total del procesamiento y la privacidad, manteniendo el costo por debajo de los 300 USD.

Flujo de trabajo típico.

1. **Adquisición.** Colocación rápida del electrodo o contacto con la piel; muestreo de 200–500 Hz para preservar el complejo QRS.
2. **Preprocesamiento.** Eliminación de deriva de línea base y ruido de red; detección robusta de picos R.
3. **Cálculo de características.** RR/BPM, PR, QRS, QT/QTc y marcadores simples de isquemia (segmento ST) cuando la calidad lo permite.
4. **Dictamen preliminar.** Reglas claras (p. ej., “RR < 0,6 s \Rightarrow taquicardia”) o modelos explicables.
5. **Registro/derivación.** Almacenamiento local y, si procede, envío seguro para revisión especializada.

Criterios técnicos mínimos para uso clínico básico.

- Frecuencia de muestreo ≥ 200 Hz y resolución efectiva ≥ 12 bits.
- Filtros en banda cardíaca y *notch* a 50/60 Hz; preservación de fase (cero-fase) para no desplazar el QRS.
- Contacto electrodo–piel estable; impedancia baja; guías claras de colocación.
- Reportes comprensibles, con marcas de R y tiempos en milisegundos.

Tabla 2.2: Comparación práctica de categorías portátiles de ECG (valores orientativos).

Categoría	Canales	Duración típica	Fortalezas	Límites
Hand-held	1–2	30 s	Fácil uso, reporte inmediato	Artefactos
Smartwatch/parche	1	Minutos–días	Continuidad, adherencia	Análisis limitado
Accesorio <i>smartphone</i>	1	Segundos–minutos	Portabilidad, bajo costo	Dependencia de dispositivo
Placa abierta (OpenBCI)	1–8	En tiempo real	Señal cruda, control total, bajo costo	Requiere procesamiento

Limitaciones comunes.

- **Cobertura diagnóstica.** Con 1–2 derivaciones no es posible localizar un infarto de forma precisa ni estimar el eje con fiabilidad.
- **Validación heterogénea.** Algunos algoritmos se entrenan con bases pequeñas o población sana, lo que reduce la generalización.
- **Interoperabilidad.** Pocas soluciones implementan estándares (HL7/FHIR) para integrarse a expedientes clínicos.
- **Privacidad.** Si el procesamiento depende de la nube, se requieren políticas y cifrado consistentes de extremo a extremo.

Ejemplos de aplicación.

- Tamizajes comunitarios y *triage* en zonas rurales.
- Correlación síntoma–evento en palpitaciones esporádicas.
- Telemedicina con alertas y segundas opiniones en tiempo casi real.

Resumen comparativo (orientativo). **Alineación con esta tesis.** El prototipo desarrollado adopta un enfoque abierto (OpenBCI Ganglion + Python/BrainFlow) y añade un clasificador basado en árboles de decisión interpretables (CART) que muestra reglas y características visibles para el clínico. De este modo se preserva la transparencia y se reduce el costo total de propiedad, manteniendo la posibilidad de auditoría y mejora continua.

Retos y oportunidades. A día de hoy existe un “vacío intermedio” entre:

- **Equipos fijos y costosos** (\$10 000–30 000 USD), que requieren infraestructura hospitalaria y personal especializado.
- **Dispositivos ligeros pero limitados** (\$100–500 USD), con una o dos derivaciones y algoritmos de “caja negra”.

Las oportunidades concretas para cerrar esa brecha son:

1. **Interpretabilidad por diseño.** Sustituir *black-box* por reglas auditables: umbrales de RR, PR, QRS, QTc y ST con justificación clínica.
2. **Procesamiento en el borde (*edge*).** Inferencia local en milisegundos para operar sin Internet; sincronización segura cuando haya conectividad.
3. **Coste total bajo.** Hardware \leq \$300 USD, software libre y hosting compartido; mantenimiento centrado en software.
4. **Integración práctica.** Exportación de reportes (PDF/CSV), API simple y compatibilidad futura con HL7/FHIR.
5. **Flujo de trabajo simple.** Captura, verificación visual (marcas R), características, dictamen preliminar y envío opcional al especialista.

Contribución de esta tesis.

- **Arquitectura híbrida *edge*–servidor:** filtrado y clasificador basado en árboles en la computadora local; respaldo y visualización web (PHP/MySQL + Plotly) para consulta y seguimiento.
- **Modelo explicable:** árbol de decisión con reglas clínicas y umbral de decisión optimizado, alineado con las características reportadas en el sistema.
- **Coste y despliegue compatibles** con brigadas de primer nivel y escuelas de medicina: un ordenador portátil común y la placa OpenBCI.

Capítulo 3

Procesamiento de señales ECG y extracción de características

En este capítulo se describe, de forma clara y paso a paso, el recorrido completo que sigue la señal electrocardiográfica: desde que se coloca un electrodo sobre la piel del paciente hasta que se obtiene un conjunto de números que pueden ser interpretados por un médico o utilizados por un algoritmo de aprendizaje automático.

Primero se explica cómo se adquiere la señal con la tarjeta *OpenBCI Ganglion*, después cómo se limpia el trazo para que se parezca a lo que mostraría un electrocardiograma clínico, y finalmente cómo se identifican las ondas P, QRS y T para calcular los intervalos de interés. A partir de estas medidas se construye un vector de características que se utilizará más adelante por el clasificador basado en árboles de decisión (Cap. 4).

La intención no es sustituir al electrocardiograma de 12 derivaciones, sino mostrar cómo un conjunto de herramientas abiertas y de bajo costo permiten acercar un apoyo diagnóstico razonable a escenarios donde, de otro modo, no habría ninguna medición objetiva de la actividad eléctrica del corazón.

3.1. Adquisición y configuración del hardware

Para registrar la señal ECG se utilizó la placa *OpenBCI Ganglion*, un dispositivo de hardware abierto con cuatro canales analógicos y resolución de 24 bit. En términos sencillos, esto significa que la placa es capaz de distinguir cambios muy pequeños de voltaje en la piel (del orden de los microvoltios), algo indispensable cuando se trabaja con señales bioeléctricas como el ECG. La comunicación con la computadora se realiza mediante Bluetooth de baja energía (BLE), lo que permite mantener al paciente sin cables rígidos y facilita su movilidad.

Esta tarjeta fue adquirida por el Posgrado en Computación de la Facultad de Ingeniería de la Universidad Autónoma de San Luis Potosí para fines educativos y para apoyar el desarrollo de proyectos de instrumentación biomédica como el que se presenta en esta tesis.¹

La captura de datos se lleva a cabo en Python utilizando la biblioteca *BrainFlow* [35, 36], desarrollada originalmente por Andrey Parfenov y colaboradores como un proyecto de código abierto orientado a la adquisición de señales neurofisiológicas. En lugar de que el usuario tenga

¹<https://docs.openbci.com/Ganglion/GanglionLanding/>

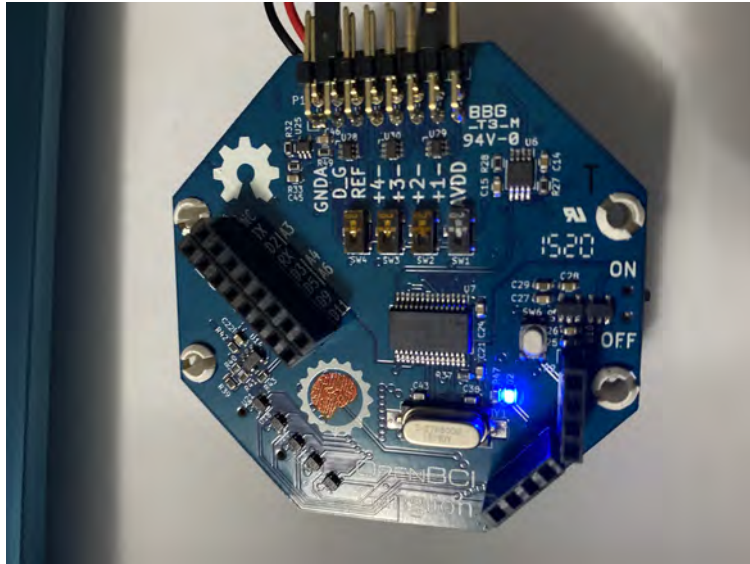


Figura 3.1: Placa *OpenBCI Ganglion* empleada en el prototipo desarrollado.

que preocuparse por detalles eléctricos o de comunicación, *BrainFlow* proporciona funciones directas para preparar la sesión, iniciar el flujo de datos y recibir la señal en forma de arreglos numéricos que pueden procesarse posteriormente.

Componentes del sistema

Componentes del sistema. El montaje físico necesario para la adquisición se compone de:

- **Placa OpenBCI Ganglion:** tarjeta principal que digitaliza la señal ECG.
- **Dongle BLED112:** adaptador USB que permite a la computadora comunicarse por Bluetooth con la placa.
- **Fuente de energía:** batería recargable, que aporta seguridad eléctrica al estar aislada de la red.
- **Electrodos adhesivos tipo snap:** se colocan sobre la piel del paciente y captan la actividad eléctrica.
- **Cables estándar:** conectan cada electrodo con la entrada correspondiente en la placa.

En este trabajo se empleó un solo canal para obtener una derivación equivalente a las derivaciones clásicas de extremidades. Para ello se utilizan dos electrodos “activos” y un tercer punto de referencia denominado D_G, que actúa como tierra o referencia común.

Limitaciones técnicas y científicas

- **Una sola derivación.** El sistema utiliza una única derivación, lo cual impide evaluar con precisión el eje eléctrico del corazón y ciertos fenómenos regionales sutiles. Algunas patologías requieren de 3 a 12 derivaciones para ser identificadas adecuadamente.
- **Tamaño y sesgo muestral.** Una parte importante de los casos anormales proviene de un solo centro (el Hospital Central de San Luis Potosí). Esto puede introducir sesgos en los tipos de pacientes y patologías observados. Se requiere una validación multisede para mejorar la generalización.
- **Variabilidad entre sujetos.** Existen diferencias anatómicas y fisiológicas entre personas (por ejemplo, en la posición del corazón, masa muscular, etc.) que afectan las amplitudes y componentes de alta frecuencia de la señal. La normalización por sujeto o por sesión podría ayudar a compensar esta variabilidad y mejorar la capacidad de generalizar.
- **Variación en las etiquetas clínicas.** La lectura de un ECG suele implicar cierto grado de interpretación por parte del especialista. La variabilidad entre observadores en el campo `diagnostico_doctor` añade ruido a los datos de entrenamiento. En un escenario ideal, habría al menos dos lecturas independientes y un consenso.
- **Artefactos y condiciones límite.** El movimiento del paciente, la actividad muscular (EMG) y el zumbido residual de la red eléctrica pueden degradar la precisión del sistema, especialmente en ritmos irregulares o de baja amplitud. Aunque se aplican filtros, los casos más extremos siguen siendo un reto.

Colocación sugerida de los electrodos

Colocación sugerida de los electrodos. Se eligió una configuración tipo derivación II, ampliamente utilizada en la práctica clínica por la claridad con la que muestra la onda P y el complejo QRS. En el sistema clásico de Einthoven, esta derivación se obtiene midiendo la diferencia de potencial entre el brazo derecho (RA) y la pierna izquierda (LL), utilizando un cuarto electrodo de referencia o tierra en la pierna derecha (RL).

En este prototipo se emplea una versión *modificada* de esa configuración: en lugar de colocar los electrodos directamente en las extremidades, se sitúan en el tórax, cerca de la proyección anatómica de cada miembro (Figura 3.2). Este desplazamiento hacia el pecho mantiene prácticamente el mismo eje eléctrico de la derivación II, pero reduce el movimiento de los cables, mejora la comodidad del paciente y disminuye el ruido por artefactos de movimiento, algo especialmente útil cuando se trabaja con un solo canal y hardware portátil.

- Electrodo negativo (RA) bajo la clavícula derecha, próximo al hombro, dentro del marco costal.
- Electrodo positivo (LL) en el hemitorax izquierdo, por debajo de los músculos pectorales y cerca del reborde costal.
- Electrodo de tierra o referencia (RL) en la pierna derecha o en su proyección abdominal inferior, conectado a la entrada `D_G` de la placa.

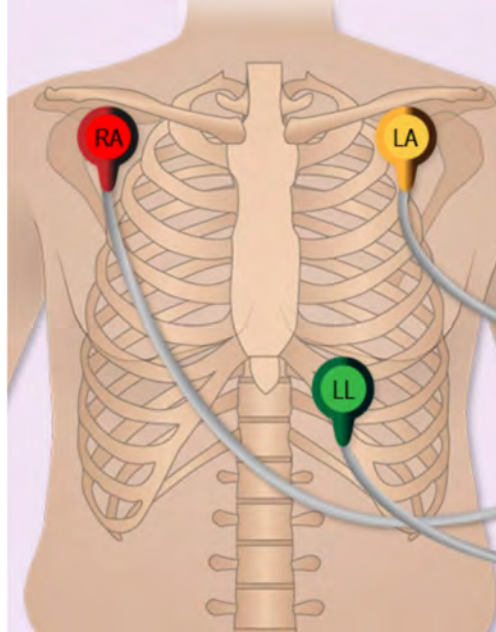


Figura 3.2: Esquema ilustrativo de la colocación de los electrodos RA, LA y LL en el tórax para obtener una derivación equivalente a las derivaciones de las extremidades. En este trabajo se utilizan RA y LL como polos de la derivación II, junto con un electrodo de referencia (RL) conectado a D_G.

Desde la perspectiva del paciente, el procedimiento consiste en limpiar los puntos de contacto con una gasa y alcohol isopropílico y adherir los parches de los electrodos. Esta limpieza sencilla reduce la impedancia de contacto (la “resistencia” entre piel y electrodo) y mejora la relación señal–ruido (SNR), es decir, ayuda a que el trazo del ECG se vea más nítido y menos contaminado por interferencias.

Esta derivación tipo II ofrece una morfología estable y fácil de interpretar, suficiente para estimar ritmo cardíaco e intervalos temporales básicos (RR, PR, QRS, QT) y para alimentar el clasificador con información relevante. Por esa razón es la derivación de referencia en la mayoría de los monitores de ritmo y Holter, y muchos dispositivos portátiles de un solo canal adoptan configuraciones equivalentes a las derivaciones de las extremidades (I o II) cuando registran el ECG a partir de pocos electrodos.

Secuencia básica con BrainFlow

Secuencia básica con BrainFlow. La lógica de adquisición en Python sigue siempre la misma secuencia, que se resume del modo siguiente:

1. Configurar un objeto `BrainFlowInputParams` con la `mac_address` de la placa y el puerto asociado al dongle BLED112.
2. Crear un objeto `BoardShim` y ejecutar `prepare_session()` para inicializar la conexión.
3. Iniciar el flujo de datos con `start_stream()`, tras lo cual la placa comienza a enviar muestras a 200 Hz (doscientas muestras por segundo).

4. Durante la captura, el programa va leyendo bloques de datos desde el búfer interno y los muestra en tiempo real en la interfaz gráfica.
5. Al finalizar, se detiene el flujo con `stop_stream()` y se libera la sesión con `release_session()`, asegurando que la placa quede lista para un nuevo estudio.

Esta secuencia es la base de todo el sistema: a partir de ella se construyen las funciones que guardan los archivos, aplican filtros, calculan los intervalos y finalmente envían los resultados al servidor web para su revisión clínica.

3.2. Parámetros de muestreo y criterios de calidad

Antes de aplicar cualquier filtro o algoritmo de detección, es necesario garantizar que la señal capturada cumple ciertos criterios mínimos de calidad. De lo contrario, el sistema podría tomar decisiones basadas en ruido en lugar de en la actividad eléctrica real del corazón.

Frecuencia de muestreo

Frecuencia de muestreo. En este prototipo se utiliza una frecuencia de muestreo de 200 Hz, lo que significa que la placa toma doscientas muestras por segundo. Este valor es suficiente para capturar con fidelidad las partes más importantes del ECG:

- Las ondas P y T, más suaves y de menor frecuencia.
- El complejo QRS, que es muy rápido y concentra energía en frecuencias más altas.

Como idea intuitiva, puede pensarse en la frecuencia de muestreo como en el número de “fotografías” que se toman por segundo de la señal. Si se toman muy pocas, el trazo se ve “cuadrulado” y se pierden detalles. Con 200 Hz se obtiene un compromiso adecuado entre detalle y cantidad de datos a procesar en tiempo real.

Resolución del conversor analógico–digital

Resolución del conversor analógico–digital. La Ganglion ofrece una resolución de 24 bits, es decir, puede dividir el rango completo de voltajes en más de 16 millones de niveles discretos. En la práctica, esto permite distinguir variaciones muy pequeñas de amplitud en la señal ECG. Para el clínico, esto se traduce en un trazo más parecido al de un electrocardiógrafo estándar, donde se aprecian con claridad las diferencias de altura entre las ondas y pequeños cambios en el segmento ST.

Criterios mínimos de calidad de señal

Criterios mínimos de calidad de señal. Para considerar que una señal es utilizable desde el punto de vista clínico, se establecen los siguientes criterios:

- **Relación señal–ruido (SNR) mayor a 10 dB:** la amplitud del ECG debe ser claramente superior al ruido de fondo. Si el ruido es comparable o mayor, la detección automática de picos R y el cálculo de intervalos se vuelven poco fiables.
- **Deriva de línea base baja:** la línea base (el nivel “cero” del trazo) no debe desplazarse de manera exagerada hacia arriba o hacia abajo, ya que esto puede ocultar el segmento ST o deformar las ondas T.

3.3. filtrado digital

Criterios de filtrado

Zumbido de red atenuado. La interferencia de la red eléctrica (60 Hz en México) debe estar suficientemente reducida mediante filtrado *notch*, es decir, un filtro digital de rechazo de banda centrado en la frecuencia de la red (50/60 Hz) que atenúa selectivamente ese componente manteniendo el resto del espectro de la señal [33,34]. De este modo, el trazo no presenta un ruido periódico que pueda confundirse con actividad cardíaca.

En la implementación práctica, estos criterios se reflejan en el diseño de los filtros digitales que se describen en las siguientes secciones: se aplica un filtrado pasa–banda para eliminar componentes muy lentos y muy rápidos que no forman parte del ECG de interés, y un filtrado pasa–stop (*notch*) para suprimir el zumbido de la red eléctrica.

Cuando la señal no cumple con estos requerimientos (por ejemplo, si algún electrodo está mal pegado y hay mucho movimiento del paciente), el sistema puede seguir graficando, pero el resultado del clasificador y las medidas numéricas deben interpretarse con cautela. Por ello, la propuesta combina siempre el cálculo automático con la posibilidad de que el médico visualice la traza filtrada y tome la decisión final, manteniendo el componente humano al centro del proceso diagnóstico.

3.4. Preprocesamiento y filtrado digital

Como se comentó en la sección anterior, la placa *OpenBCI Ganglion* es un dispositivo de adquisición biomédica portátil basado en Bluetooth Low Energy, capaz de muestrear hasta cuatro canales de ECG a 200 Hz con una resolución de 24 bit por canal. Gracias a su *SDK* abierto y a la biblioteca *BrainFlow* para Python, la señal cruda puede capturarse con unas pocas llamadas a la API y almacenarse directamente en arreglos numéricos de alto nivel listos para su procesamiento [35,36].

La señal de ECG que llega directamente de los electrodos no es perfecta: suele presentar deriva de línea de base, interferencia de la red eléctrica y ruido muscular o de movimiento. Si estas componentes no se atenúan, el sistema podría equivocarse al localizar los picos R o al medir los intervalos PR, QRS y QT. Por ello se aplica un *pipeline* de preprocesado en cuatro etapas, que se ejecuta en doble pasada (*zero–phase*) utilizando las funciones de filtrado incluidas en *BrainFlow* [33]:

1. **Eliminación de componente DC (*detrending*).** En una primera etapa se elimina la componente de continua (DC) mediante la función `DataFilter.detrend(signal,`

`DetrendOperations.CONSTANT`). En términos sencillos, esto equivale a “recentrar” el trazo alrededor de cero, suprimiendo el desplazamiento lento de la línea de base producido por la respiración, cambios posturales o variaciones en el contacto de los electrodos.

2. **Filtro pasa–banda Butterworth de 4.º orden, 0.5 Hz a 45 Hz.** A continuación se aplica un filtro pasa–banda que conserva la banda de interés clínico donde se encuentra la mayor parte de la energía de las ondas P, del complejo QRS y de la onda T. Este filtro atenúa:
 - componentes de muy baja frecuencia ($< 0,5$ Hz), asociadas a la deriva de línea de base;
 - componentes de alta frecuencia (> 45 Hz), relacionadas con ruido muscular o interferencias rápidas.

El filtro Butterworth de 4.º orden ofrece una respuesta suave en la banda de paso, lo que ayuda a conservar la forma característica de las ondas del ECG sin introducir ondulaciones artificiales.

3. **Filtro *notch* Butterworth de 3.º orden, centro 60 Hz (Q=35).** En México, la red eléctrica opera a 60 Hz. Para eliminar este “zumbido” se utiliza un filtro *notch* (pasa–rechaza) centrado en 60 Hz, con un factor de calidad $Q = 35$, tal como se recomienda en el módulo de filtrado de *BrainFlow* [33]. Este filtro abre un “hueco” muy estrecho alrededor de esa frecuencia, reduciendo de forma selectiva la interferencia de la red sin afectar el contenido útil del ECG.
4. **Estandarización tipo *z–score*.** Finalmente, se normaliza la señal filtrada aplicando la transformación

$$z = \frac{x - \mu}{\sigma},$$

donde μ es la media y σ la desviación estándar de la señal. El resultado es un trazo donde la mayor parte de los valores oscila en torno a cero y con una escala comparable entre diferentes pacientes y registros. Esta estandarización permite fijar umbrales de detección más robustos; por ejemplo, seleccionar como candidatos a pico R aquellos valores que superan un cierto porcentaje (por ejemplo, el 35 % del valor máximo) o un múltiplo de la desviación estándar.

En segundo plano, *BrainFlow* se encarga de desempaquetar el protocolo de datos del Ganguion (que agrupa cinco muestras por envío BLE) y de presentarlos como arreglos de `float64`. Desde el punto de vista del programa, aplicar los filtros anteriores resulta tan sencillo como invocar métodos como `BoardShim.start_stream()`, `DataFilter.perform_bandpass()`, `DataFilter.perform_bandstop()` y `DataFilter.detrend()`, siguiendo los ejemplos oficiales [37].

Con este preprocesamiento se garantiza que la señal resultante conserva la morfología cardíaca esencial (ondas P bien definidas, complejos QRS nítidos y ondas T reconocibles) y que los ruidos de línea y musculares quedan suficientemente atenuados. Esta señal limpia es la que se utiliza en la etapa siguiente para detectar automáticamente los picos y calcular los intervalos clínicos, como se describe en la Sección 3.5.

3.5. Detección de ondas y puntos característicos

Una vez obtenida la señal ECG preprocesada, el siguiente paso es localizar los hitos electrocardiográficos clave: picos R (complejo QRS), puntos Q y S, y las ondas P y T. Para ello se utiliza un algoritmo derivado de Pan–Tompkins, optimizado para ejecución en tiempo real y resiliente a pequeñas variaciones de amplitud o ruido residual [38].

Detección de picos R (QRS). El método comienza con la aplicación de la derivada digital, seguida de una elevación al cuadrado de la señal para enfatizar los picos y suprimir componentes de baja amplitud. A continuación se realiza una integración móvil que promueve un único máximo por latido. Finalmente, se impone una distancia mínima de 300 ms entre picos consecutivos para evitar falsos positivos en ritmos rápidos o en presencia de artefactos [39].

Localización de ondas P y T. Para cada pico R detectado, se define una ventana anterior de 120 ms a 200 ms en la cual se busca el máximo local correspondiente a la onda P, y una ventana posterior de 150 ms a 400 ms donde se localiza el máximo que corresponde a la onda T. Estos rangos temporales se basan en la fisiología normal del ciclo cardíaco y permiten adaptarse a frecuencias cardíacas entre 40 y 150 latidos por minuto [34].

Identificación de puntos Q y S. Los puntos Q y S se encuentran buscando el mínimo local justo antes y justo después del pico R, respectivamente, dentro de un radio de 60 ms. Esta estrategia garantiza que los complejos QRS anchos (p.ej. en bloqueos de rama) sigan siendo detectados correctamente, ya que el radio de búsqueda cubre la mayor parte de la duración del QRS patológico [40].

Con las latencias y amplitudes de estos cinco puntos definidos por latido, es posible calcular los intervalos clínicos fundamentales:

$$\begin{aligned} RR &= t_{R_{i+1}} - t_{R_i}, \\ PR &= t_{R_i} - t_{P_i}, \\ QRS &= t_{S_i} - t_{Q_i}, \\ QT &= t_{T_i} - t_{Q_i}. \end{aligned}$$

Estos parámetros forman la base de los análisis morfológicos y del vector de características empleado para la clasificación de ritmos normales y patológicos.

3.6. Extracción de características

Una vez localizados los hitos electrocardiográficos (picos R, puntos Q y S, y las ondas P y T), el siguiente paso consiste en traducir la forma del trazo en números. Es decir, se pasa de “ver” un electrocardiograma a “medirlo” de manera sistemática.

Estas medidas describen tanto la dinámica temporal (cuánto tarda el corazón en completar cada fase del ciclo) como ciertos aspectos básicos de la morfología (relaciones entre amplitudes). A partir de ellas se construye un vector de características compacto que:

- es suficientemente pequeño para ser manejado por modelos de clasificación ligeros (como los árboles de decisión tipo CART);
- conserva información clínicamente interpretable, de forma que un cardiólogo pueda relacionar cada número con un fenómeno fisiológico conocido.

En otras palabras, el vector de características funciona como una “ficha resumida” de la actividad eléctrica del corazón durante la ventana de registro. Cada registro de ECG aporta las siguientes variables:

- **Intervalo RR (variabilidad del ritmo).** Se calculan la media, la desviación típica y la RMSSD (Root Mean Square of Successive Differences) de los intervalos RR. Estos tres valores describen cómo varía el tiempo entre un latido y el siguiente:
 - La *media* del RR indica el ritmo global (corazón más bien lento, normal o rápido).
 - La *desviación típica* muestra qué tanto se dispersan los intervalos alrededor de ese valor medio.
 - La *RMSSD* mide las diferencias de un latido al siguiente y es especialmente sensible a las fluctuaciones de corto plazo.

En términos clínicos, una RMSSD elevada puede ser un marcador temprano de arritmias supraventriculares o de ritmos muy irregulares, mientras que una variabilidad extremadamente baja puede asociarse a situaciones de estrés o a disfunción del sistema nervioso autónomo.

- **Duración de segmentos (PR, QRS, QT y ST).** Se miden las duraciones medias de los intervalos PR, QRS, QT y ST. Cada uno refleja un fenómeno diferente:
 - **PR:** tiempo de conducción desde las aurículas hasta los ventrículos.
 - **QRS:** duración de la despolarización ventricular.
 - **QT:** tiempo total de actividad eléctrica ventricular (despolarización + repolarización).
 - **ST:** fase inicial de la repolarización ventricular.

Desviaciones respecto a rangos de referencia (por ejemplo, PR > 200 ms, QRS > 120 ms, QT corregido > 450 ms) se han asociado clásicamente a bloqueo auriculoventricular, bloqueo de rama o prolongación del QT, respectivamente. Incluir estas duraciones en el vector de características permite que el modelo de aprendizaje automático “detecte” estos patrones de manera automatizada.

- **Índice QTc (QT corregido).** Para tener en cuenta el efecto de la frecuencia cardiaca sobre la duración del QT, se calcula el QT corregido mediante la fórmula de Bazett:

$$QTc = \frac{QT}{\sqrt{RR}}. \quad (3.1)$$

Este índice permite comparar valores de QT entre pacientes con ritmos cardiacos diferentes. Por ejemplo, un QT aparentemente normal en un paciente con frecuencia baja puede ser, en realidad, prolongado cuando se corrige por el intervalo RR.

- **Relaciones de amplitud (R/T y R/S).** Además de los tiempos, se consideran relaciones simples entre amplitudes:
 - **R/T:** cociente entre la amplitud del pico R y la amplitud de la onda T.
 - **R/S:** cociente entre la amplitud del pico R y la amplitud del punto S.

Estas proporciones ayudan a caracterizar la forma del complejo QRS y de la onda T. En la práctica clínica, ciertos patrones (por ejemplo, un valor de R/T muy elevado o una R/S anómala) se relacionan con hipertrofias ventriculares, sobrecarga de cavidades o alteraciones en la recuperación eléctrica (isquemia, cambios en el segmento ST).

- **Frecuencia cardiaca instantánea (BPM).** Finalmente, se incluye la frecuencia cardiaca expresada en latidos por minuto (BPM), calculada como

$$\text{BPM} = \frac{60}{RR}. \quad (3.2)$$

Este valor resume el ritmo global durante la ventana analizada. Más allá de indicar si el corazón late lento, normal o rápido, su combinación con la variabilidad del RR puede aportar pistas sobre respuestas al ejercicio, al estrés o sobre la presencia de disautonomías.

De esta manera, cada registro queda descrito por un conjunto reducido de características que conservan un significado clínico claro y, al mismo tiempo, resultan manejables para el clasificador que se describe en el capítulo siguiente.

Frecuencia cardiaca promedio (BPM) a partir del intervalo RR

La frecuencia cardiaca promedio se obtiene directamente a partir de los intervalos T_{RR} , que representan la diferencia de tiempo entre dos picos R consecutivos. En la práctica, el sistema primero calcula el tiempo medio que transcurre entre latido y latido y, después, lo convierte a latidos por minuto.

1) Definición de BPM. La frecuencia cardiaca promedio se define como

$$\overline{\text{BPM}} = \frac{60}{\overline{T_{RR}}}, \quad (3.3)$$

donde $\overline{T_{RR}}$ se expresa en segundos. De forma intuitiva, si el intervalo entre latidos es largo (corazón lento), la BPM será baja; si el intervalo es corto (corazón rápido), la BPM será alta.

2) Promedio de los intervalos RR. El valor $\overline{T_{RR}}$ se calcula como el promedio de todos los intervalos RR detectados:

$$\overline{T_{RR}} = \frac{1}{N_R} \sum_{n=1}^{N_R} (t_{R(n+1)} - t_{R(n)}), \quad (3.4)$$

donde $t_{R(n)}$ es la marca temporal (en segundos) del n -ésimo pico R, y N_R es el número total de intervalos considerados. En otras palabras, se suman todos los tiempos entre latido y latido y se divide entre el número de intervalos.

3) Equivalencia con BPS. Si se divide $\overline{\text{BPM}}$ entre 60 se obtiene el número de latidos por segundo (BPS, por sus siglas en inglés). Aunque en clínica se usa casi siempre el valor en BPM, la forma en BPS puede ser útil en ciertos análisis de señal.

Uso en el *pipeline*. En el sistema propuesto, una vez detectados los picos R y calculadas las marcas temporales $t_{R(n)}$, se obtiene \overline{T}_{RR} mediante la ecuación (3.4) y, de inmediato, la frecuencia cardiaca promedio $\overline{\text{BPM}}$ con la ecuación (3.3). Ambos valores (intervalo RR medio y BPM) se integran al vector de características descrito en esta sección, de modo que el clasificador no solo conozca la forma de cada latido, sino también el contexto del ritmo global en el que se producen.

3.7. Control de artefactos y criterios de exclusión

En la práctica, ningún registro de ECG está libre de imperfecciones. Aunque se cuente con una buena colocación de electrodos y un equipo adecuado, siempre aparecen señales no deseadas que pueden confundir tanto al especialista como al algoritmo de clasificación. Estas señales parasitarias se conocen como *artefactos* y es importante reconocerlas y controlarlas desde el propio diseño del sistema.

En el caso del prototipo desarrollado, los artefactos más frecuentes son:

- **Interferencia de red eléctrica (60 Hz) y ruido electromiográfico (EMG).** Corresponden al “zumbido” producido por la instalación eléctrica y a las señales generadas por los músculos cercanos (por ejemplo, si el paciente tensa las piernas o los brazos).
- **Deriva de la línea de base.** Se observa como un desplazamiento lento de todo el trazo hacia arriba o hacia abajo, normalmente asociado a la respiración, a movimientos del paciente o a cambios en el contacto de los electrodos.
- **Artefactos de movimiento o contacto deficiente.** Se manifiestan como picos bruscos, saltos repentinos o segmentos “aplanados” cuando un electrodo se despegó parcialmente.

El sistema propuesto aborda estos problemas combinando el filtrado digital con criterios prácticos de calidad de señal:

Interferencia de red y EMG. La interferencia de la red eléctrica se atenúa mediante un filtro *notch* centrado en 60 Hz, complementado con el filtro pasa-banda de 0.5 Hz a 45 Hz descrito en la Sección 5. De esta forma, la mayor parte del zumbido eléctrico y del ruido muscular de alta frecuencia se reduce antes de la detección de picos R. En términos sencillos, el sistema “limpia” la señal para que se parezca lo más posible al trazo que se observaría en un electrocardiograma clínico.

Deriva de línea de base. La deriva se corrige aplicando técnicas de *detrending* (eliminación de la componente de continua) y, en caso necesario, un filtro pasa-alto suave que elimina variaciones muy lentas. El objetivo es que el nivel cero de referencia se mantenga estable, evitando que el segmento ST quede oculto o deformado por un desplazamiento lento del trazo.

Criterios prácticos de exclusión. A pesar de estos mecanismos de corrección, hay situaciones en las que la calidad de la señal no es suficiente para confiar en los resultados automáticos. Para ello se establece un conjunto de criterios simples:

- Relación señal–ruido (SNR) menor a 10 dB.
- Zumbido residual de red claramente visible tras el filtrado.
- Deriva de línea de base marcada o artefactos de movimiento frecuentes.

Cuando se presenta alguna de estas condiciones, el registro se marca como de *calidad deficiente*. En el flujo de trabajo, estos registros pueden:

- Excluirse del conjunto de entrenamiento del modelo de aprendizaje automático, para evitar que el clasificador “aprenda” patrones que en realidad corresponden a ruido.
- Mantenerse únicamente para fines de prueba de robustez, pero no para el cálculo de características finales.

Este control de calidad combina la visión técnica (filtros y SNR) con una mirada clínica: se prefiere sacrificar algunos registros dudosos antes que construir un sistema que tome decisiones basadas en trazos poco confiables.

3.8. Formato de archivos y persistencia

Una vez que termina cada sesión de captura y filtrado, el sistema no solo muestra el ECG en pantalla, sino que también guarda de forma organizada toda la información necesaria para su análisis posterior. La idea es que, si un médico desea revisar un estudio semanas o meses después, pueda encontrar de manera sencilla tanto el trazado como los datos numéricos y el contexto clínico asociado.

Para ello, cada registro se almacena en tres niveles complementarios:

- **Archivo CSV filtrado.** Por cada sesión se genera un archivo en formato CSV (*Comma Separated Values*) que contiene la señal ECG ya filtrada. En este trabajo se utiliza una sola columna (un canal) muestreada a 200 Hz durante 60 s, lo que da lugar a unas 12 000 muestras por archivo. Este formato tabular es sencillo de abrir tanto con herramientas de análisis (Python, MATLAB, R) como con hojas de cálculo, y facilita la lectura programática, la depuración y la auditoría de los datos. En caso de ser necesario, otro investigador puede descargar el CSV y repetir el preprocesamiento o probar nuevos algoritmos sobre el mismo material.
- **Imagen PNG con cuadrícula clínica.** Además del CSV, el sistema genera una imagen en formato PNG que muestra el trazo ECG sobre una malla similar a la utilizada en los electrocardiogramas convencionales. La cuadrícula se configura con pasos equivalentes a 1 mV en el eje vertical y 200 ms en el horizontal, de acuerdo con la frecuencia de muestreo utilizada. Esta representación visual resulta familiar para el personal clínico, ya que recuerda al papel milimetrado de un ECG tradicional. Facilita la revisión rápida, la toma de notas manuales (por ejemplo, impresas sobre papel) y la explicación al paciente, integrando la tecnología digital con la experiencia cotidiana del consultorio.

- **Metadatos en base de datos (BD).** Finalmente, toda la información contextual y las medidas derivadas se almacenan en una base de datos MySQL. Para cada registro se guardan, entre otros:
 - Datos básicos del paciente: edad, sexo y fecha/hora del estudio.
 - características calculadas: intervalos RR, PR, QRS, QT, ST, frecuencia cardíaca (BPM), QTc, entre otros.
 - Rutas a los archivos CSV y PNG generados.
 - Un identificador único (*folio*) que vincula todos estos elementos.

De forma opcional, se registra también un hash criptográfico (por ejemplo, SHA-256) asociado a los archivos, lo que permite comprobar su integridad y asegurar que no han sido modificados desde el momento de la captura. Este detalle es relevante cuando se piensa en futuras aplicaciones médico-legales o en estudios de investigación donde se requiere trazabilidad robusta de los datos.

En conjunto, estos tres niveles de almacenamiento (señal numérica, representación gráfica y metadatos clínicos) permiten construir una base de datos coherente, reutilizable y transparente, en la que cada registro puede ser rastreado desde el archivo original hasta las decisiones del clasificador.

3.9. Protocolo de adquisición de la base de datos

Para entrenar y evaluar los clasificadores descritos en capítulos posteriores fue necesario contar con una base de datos propia de señales ECG capturadas con el sistema desarrollado. El objetivo no fue competir en tamaño con grandes bases públicas, sino disponer de un conjunto de registros cuidadosamente controlados y bien documentados, que reflejen el contexto real de uso del prototipo.

Se recopiló un conjunto de 100 registros de ECG en derivación II, divididos equitativamente en 50 muestras de sujetos sin patología cardiovascular conocida y 50 de pacientes con diagnóstico confirmado de cardiopatía (incluyendo arritmias, isquemia miocárdica o bloqueos de conducción). El procedimiento se llevó a cabo siguiendo normas clínicas rigurosas y contó con la aprobación del comité de ética correspondiente (folio #ECG25-01). En todo momento se respetaron los principios de confidencialidad y consentimiento informado.

Preparación y posicionamiento. Antes de cada registro se siguió un protocolo estandarizado:

- Los voluntarios se colocaron en reposo, en decúbito supino (acostados boca arriba), respirando de manera espontánea y procurando evitar movimientos bruscos.
- Se limpió la piel en los puntos de contacto con una gasa y alcohol isopropílico, con el fin de reducir la impedancia y mejorar la calidad de la señal.
- Se colocaron los electrodos de acuerdo con la derivación II: electrodo negativo en brazo derecho (RA), electrodo positivo en pierna izquierda (LL) y electrodo de tierra en pierna derecha (RL), conectados a la placa Ganglion como se describió en la Sección 3.1.

Esta preparación busca reproducir las condiciones en las que habitualmente se realiza un ECG clínico, pero utilizando el hardware de bajo costo del sistema propuesto.

Condiciones de registro. Las grabaciones se realizaron bajo las siguientes condiciones:

- *Duración y muestreo:* cada registro tuvo una duración de 30s y se adquirió a una frecuencia de muestreo de 200 Hz. En casos donde el cardiólogo observó variabilidad patológica relevante (por ejemplo, arritmias intermitentes), se extendió la captura hasta 300s para documentar mejor el comportamiento del ritmo.
- *Pulso de calibración:* antes y después de cada sesión se aplicó un pulso de calibración de 1 mV, que permite verificar la ganancia y la linealidad del sistema de adquisición. Este paso es análogo a la calibración que realizan los electrocardiógrafos comerciales y ofrece una referencia común al interpretar amplitudes.

Esta metodología asegura que los datos utilizados para entrenar el modelo no son simples señales “de laboratorio”, sino registros obtenidos en condiciones parecidas a las de un entorno clínico real.

Documentación y metadatos. Además del trazo ECG en sí mismo, se recabó información adicional que ayuda a contextualizar cada registro:

- *Ficha clínica básica:* edad, sexo, antecedentes relevantes (por ejemplo, hipertensión, diabetes, tabaquismo) y medicación actual, cuando ésta podía influir en la conducción o en la repolarización cardíaca.
- *Consentimiento informado:* todos los participantes firmaron un documento de consentimiento informado en el que se explica el propósito del estudio y el uso de sus datos con fines de investigación.
- *Anonimización:* cada registro se identificó mediante un `folio_paciente` que no contiene datos personales directos. De este modo, el sistema puede enlazar señales, diagnósticos y características sin exponer la identidad de los participantes.

Esta combinación de señal, contexto clínico y protección de la identidad es clave para que la base de datos pueda utilizarse con seguridad tanto en investigación como en futuras extensiones del sistema.

Colocación de electrodos para ECG de 12 derivaciones

Aunque el sistema propuesto trabaja principalmente con una sola derivación (derivación II), en la práctica clínica el electrocardiograma estándar se realiza con 12 derivaciones. Esto proporciona una visión más completa de la actividad eléctrica del corazón desde diferentes “ángulos”.

Las siguientes figuras ilustran la colocación correcta de los electrodos en extremidades y en el tórax, siguiendo el esquema clásico de 12 derivaciones. En la Fig. 3.3 se muestra la colocación en miembros inferiores y superiores para obtener las derivaciones de extremidades,

mientras que la Fig. 3.4 ilustra el montaje de las derivaciones precordiales V_1-V_6 sobre la pared torácica anterior.



Figura 3.3: Colocación de los electrodos en los miembros inferiores (izquierda) y miembros superiores (derecha) para derivaciones de extremidades. Estas posiciones permiten obtener las derivaciones I, II, III, aVR, aVL y aVF.



Figura 3.4: Montaje de las derivaciones precordiales V_1-V_6 en la pared torácica anterior, que complementan la información de las derivaciones de extremidades al mostrar la actividad eléctrica desde distintas zonas del tórax.

Estas imágenes sirven como referencia visual para el lector no familiarizado con el ECG de 12 derivaciones y permiten entender cómo la derivación II utilizada en este trabajo se integra dentro del esquema completo empleado habitualmente en cardiología.

3.10. Almacenamiento y etiquetado

Tras la captura y procesado de cada sesión, los datos y sus anotaciones se organizan de la siguiente manera:

- **CSV filtrado:** El flujo de datos genera un archivo CSV en el que cada fila representa un canal de ECG. En la versión final, únicamente se conserva la columna correspondiente a la derivación II, ya pretratada con los filtros de banda y *notch* descritos en la sección 3.4. Este formato tabular facilita la lectura programática, la reposición incremental y la auditabilidad del procesamiento de la señal.
- **PNG con cuadrícula clínica:** Para cada registro se genera una imagen en formato PNG que exhibe la traza filtrada superpuesta a una malla estándar de 1 mV en vertical por 200 ms en horizontal. Esta representación gráfica es útil para una revisión rápida por parte de personal clínico y permite la anotación manual de picos o artefactos fuera de rango.

Metadatos y base de datos:

- Todos los parámetros demográficos y clínicos (edad, sexo, diagnóstico definitivo, medicación concomitante) se almacenan en una base de datos MySQL.
- Cada nuevo registro recibe un identificador único `folio_paciente`, generado en el servidor PHP, garantizando un espacio de folios de cinco dígitos.
- En el momento de la inserción en la tabla `t_archivos` un disparador SQL calcula un valor SHA-256 sobre el contenido del CSV filtrado y lo registra en la columna `hash_sha256`.
- Durante la subida de datos (endpoint `save_data.php`) se verifica que el hash remitido por el cliente coincida con el almacenado, asegurando la integridad de la señal transmitida.

3.11. Selección del hardware *OpenBCI Ganglion*

La elección del dispositivo de adquisición de señales ECG es crítica para garantizar calidad diagnóstica, portabilidad y coste contenido. Tras evaluar varias alternativas, el *OpenBCI Ganglion* se mostró como la opción más balanceada para este proyecto gracias a sus prestaciones técnicas, su ecosistema de desarrollo abierto y su precio accesible.

Prototipo 0 – Arduino Nano + AD8232

El AD8232 es un frente analógico integrado de un solo canal, diseñado para monitores de frecuencia cardíaca portátiles. Ofrece filtros ajustables y un amplificador de alta ganancia, pero entrega una señal analógica que debe ser digitalizada por el microcontrolador. En el caso del Arduino Nano (y del Arduino Nano RP2040), la conversión A/D disponible es de

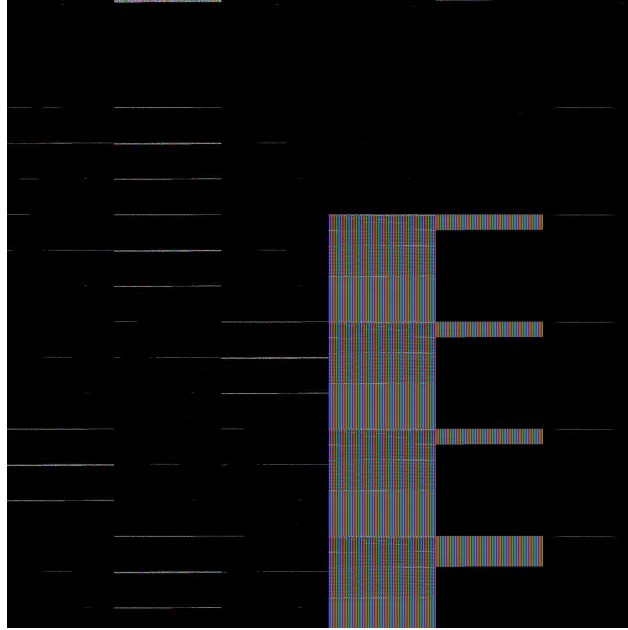


Figura 3.5: Montaje exploratorio con *Arduino Nano + AD8232*. Este prototipo permitió depurar la cadena de filtrado y la interfaz de visualización, aunque su resolución efectiva (10–12 bits en el convertidor A/D del microcontrolador) resultó limitada para uso clínico.

10–12 bits, lo que implica un número mucho menor de niveles de cuantificación que los 24 bits de la Ganglion. En la práctica, esto se traduce en menor detalle en la forma de las ondas y en una menor capacidad para distinguir cambios sutiles en el segmento ST o en la amplitud de la onda T, aspectos especialmente importantes cuando se pretende apoyar un diagnóstico clínico.

Ventajas técnicas principales de la Ganglion

Tabla 3.1: Comparación básica de alternativas exploradas.

Característica	Ganglion	AD8232	Arduino Nano RP2040
Resolución (bits)	24 (ADC dedicado)	Analógico	12 (ADC integrado)
Canales ECG	4	1	1
Rango dinámico típico	$\pm 4,5$ mV aprox.	$\pm 1,5$ mV aprox.	$\pm 1,0$ mV aprox.
Bluetooth Low Energy (BLE)	Sí	No	No (requiere módulo externo)
Firmware/SDK abiertos	Sí	Parcial	Sí (ecosistema Arduino)
Costo aproximado (USD)	< 300	< 30	< 40

Esta comparación resume por qué el prototipo basado en Arduino se consideró solo un paso previo: el coste es muy bajo, pero la resolución y el número de canales son limitados. Para un sistema cuyo objetivo es apoyar decisiones clínicas y calcular medidas sensibles a cambios sutiles (QTc, variabilidad del ritmo, morfología del ST), la Ganglion aporta una mejor relación señal–ruido y un margen de trabajo más cercano al de un equipo médico, manteniendo un precio razonable para investigación aplicada.

Comparación con la tesis de Sánchez Ovalle (2020)

En su prototipo, Sánchez Ovalle empleó el módulo AD8232 acoplado a un teléfono Android, lo que implicaba:

- *Filtrado inexistente en tiempo real*, relegando el posprocesado al servidor o a la propia aplicación móvil.
- *Dependencia de la batería del móvil* para la ganancia analógica, con variaciones de señal según el nivel de carga.
- *Canal único y baja resolución*, limitando la calidad de la señal y la robustez del diagnóstico.

Por el contrario, la *OpenBCI Ganglion* incorpora un reloj interno estable, verdadera derivación diferencial y salida digital aislada, lo que:

- Reduce el ruido de modo común.
- Asegura consistencia en la frecuencia de muestreo (200 Hz) y en los niveles de amplificación.
- Mejora la portabilidad y la autonomía de uso en entornos rurales o de atención primaria.

Comparación con *OpenBCI Cyton* y otros frentes analógicos

Existen otras plataformas de adquisición abiertas y frentes analógicos específicos para ECG que podrían haberse considerado. Entre ellas destacan:

- ***OpenBCI Cyton***. La placa *Cyton* es una plataforma de 8 canales (ampliable a 16 con un módulo *Daisy*) con resolución de 24 bits y muestreo típico de 250 Hz por canal. Está pensada principalmente para aplicaciones EEG multicanal y ofrece una gran flexibilidad en número de derivaciones y configuraciones. Sin embargo, su coste es significativamente mayor y su capacidad multicanal excede las necesidades de este proyecto, que se centra en una derivación equivalente y en un máximo de cuatro canales ECG. Para un sistema portátil orientado a un entorno clínico de recursos moderados, la *Ganglion* proporciona una configuración suficiente con menor complejidad y menor inversión inicial.
- **Frente analógico ADS1293 (Texas Instruments)**. El *ADS1293* es un frente analógico específico para ECG con tres canales de 24 bits, muy bajo consumo de potencia y características avanzadas (detección de “electrodo suelto”, *right-leg drive*, terminales de Wilson y Goldberger, entre otras). Se comercializa como circuito integrado y como módulo de evaluación. Esta solución es muy atractiva para el diseño de dispositivos comerciales dedicados, pero implica desarrollar una tarjeta electrónica propia, encargarse de la comunicación digital y de la certificación del hardware. Para un trabajo de maestría enfocado en el procesamiento de señales y en la integración con una plataforma web, esta ruta supondría un esfuerzo de ingeniería electrónica adicional que escapa al alcance planteado.

En resumen, la Ganglion se sitúa en un punto intermedio entre los módulos muy económicos de un solo canal (como el AD8232) y las plataformas multicanal de mayor coste (como *Cyton* o diseños basados en frentes analógicos dedicados). La combinación de 4 canales, 24 bits de resolución, BLE, batería y firmware abierto, junto con el soporte de bibliotecas como *BrainFlow*, hace que resulte especialmente adecuada para prototipos clínicos de bajo costo como el presentado en esta tesis.

Capítulo 4

Aprendizaje automático y árboles de decisión

En los capítulos anteriores se describió cómo se adquiere la señal electrocardiográfica, cómo se filtra el ruido y cómo se transforman los latidos en un conjunto de medidas con significado clínico (intervalos, segmentos y frecuencia cardíaca). El siguiente paso consiste en utilizar esas medidas como entrada de un modelo capaz de sugerir si un registro es compatible con un patrón *Normal* o *Anormal*.

Este tipo de problema se aborda mediante técnicas de *aprendizaje automático* supervisado. De manera informal, el aprendizaje automático estudia algoritmos que aprenden a partir de ejemplos: se les muestran muchos estudios de ECG ya etiquetados por un médico, y el modelo ajusta sus parámetros para imitar, en la medida de lo posible, esas decisiones clínicas. Una vez entrenado, el modelo puede recibir un nuevo conjunto de características y emitir una recomendación automática.

Dentro de los modelos supervisados, los *árboles de decisión* ocupan un lugar especial porque sus reglas son fáciles de interpretar. Un árbol puede pensarse como una serie de preguntas encadenadas del tipo “¿el QTc está por encima de cierto valor?” o “¿la duración del QRS es mayor que un umbral?”, hasta llegar a una conclusión final: *Normal* o *Anormal*. En esta tesis se utilizan tanto árboles individuales (modelo CART) como conjuntos de muchos árboles (*Random Forest*) para mejorar la estabilidad de las predicciones.

El objetivo de este capítulo es presentar, de forma accesible para lectores que no son especialistas en inteligencia artificial, los conceptos básicos del aprendizaje automático supervisado, el funcionamiento de los árboles de decisión y de los bosques aleatorios, así como las principales métricas con las que se evaluará su desempeño. La configuración concreta del modelo, los experimentos realizados y los resultados obtenidos con las señales ECG del prototipo se describen en las secciones siguientes.

4.1. Fundamentos del aprendizaje automático supervisado

El problema que se aborda en esta tesis puede formularse como un problema de *clasificación supervisada*. En este tipo de tareas se dispone, para cada paciente, de dos elementos:

por un lado, un conjunto de medidas numéricas extraídas del ECG (intervalos, segmentos, frecuencia cardíaca, etc.), las cuales se describen en el Capítulo 3, en las Secciones 3.5 y 3.6, así como en la sección 3.7; por otro, una etiqueta clínica asignada por el médico (por ejemplo, *Normal* o *Anormal*). El modelo de aprendizaje automático intenta aprender la relación entre ambos.

Cada estudio de ECG se representa mediante un vector de características fisiológicas y morfológicas

$$\mathbf{x} = (x_1, \dots, x_p) \in \mathbb{R}^p,$$

donde cada componente x_j puede corresponder, por ejemplo, a la duración media del complejo QRS, al valor del QT corregido (QTc) o al intervalo RR. A cada vector de características se le asocia una etiqueta binaria

$$y \in \{0, 1\},$$

donde, en este trabajo, $y = 1$ representa un registro *Anormal* y $y = 0$ un registro *Normal*.

El objetivo del modelo es aproximar la probabilidad condicional $\mathbb{P}(y = 1 \mid \mathbf{x})$, es decir, estimar qué tan probable es que un estudio con ciertas características pertenezca a la clase anormal. A partir de esa probabilidad y de un umbral de decisión t , se obtiene una clasificación binaria:

$$\hat{y} = \begin{cases} 1, & \text{si } \mathbb{P}(y = 1 \mid \mathbf{x}) \geq t, \\ 0, & \text{si } \mathbb{P}(y = 1 \mid \mathbf{x}) < t. \end{cases}$$

En términos generales, el aprendizaje supervisado sigue tres ideas básicas:

1. Entrenamiento con ejemplos etiquetados.

El modelo ajusta sus parámetros utilizando un conjunto de estudios donde tanto \mathbf{x} como y son conocidos. Durante esta fase el modelo “revisa” muchos ejemplos de ECG normales y anormales y aprende patrones que los diferencian.

2. Evaluación en ejemplos nuevos.

Para comprobar si el modelo ha aprendido algo general y no solo ha memorizado los ejemplos de entrenamiento, se mide su desempeño sobre estudios que no se usaron para entrenarlo. Este principio es la base de técnicas como la validación cruzada que se revisará más adelante.

3. Uso del modelo entrenado.

Una vez validado, el modelo puede recibir un nuevo vector de características \mathbf{x} y emitir una probabilidad de anormalidad $\mathbb{P}(y = 1 \mid \mathbf{x})$, que posteriormente se traduce a una recomendación *Normal/Anormal* mediante el umbral t .

En las secciones siguientes se describen con mayor detalle los tipos de modelos que se utilizarán (en particular, los árboles de decisión y los bosques aleatorios) y las métricas que permiten valorar su desempeño.

4.2. Partición, validación y evitación de fugas de información

Una vez definidas las variables de entrada y la etiqueta de salida del problema de clasificación, es indispensable establecer un procedimiento riguroso para evaluar el desempeño del modelo de aprendizaje automático. Esta etapa resulta crítica, ya que una evaluación inadecuada puede conducir a estimaciones excesivamente optimistas que no se reproducen cuando el sistema se enfrenta a datos verdaderamente nuevos.

En términos generales, la validación de modelos persigue estimar su capacidad de generalización, es decir, su desempeño sobre muestras no utilizadas durante el proceso de entrenamiento. Para ello, es necesario particionar adecuadamente el conjunto de datos y asegurar que la información utilizada para ajustar el modelo no se filtre, de manera directa o indirecta, hacia el conjunto de evaluación. Este fenómeno, conocido como *fuga de información* (*data leakage*), compromete la validez de los resultados y debe evitarse mediante una separación estricta entre datos de entrenamiento y prueba.

En este trabajo se adopta un esquema de validación que busca aproximarse, en la medida de lo posible, a un escenario de uso real, donde el modelo debe emitir predicciones sobre instancias que no ha observado previamente.

Validación cruzada estratificada

Para la evaluación del clasificador se emplea validación cruzada estratificada *k-fold*. Bajo este esquema, el conjunto de datos se divide en k subconjuntos o *pliegues*; en el presente estudio se utiliza $k = 8$. En cada iteración del proceso:

1. El modelo se entrena utilizando $k - 1$ pliegues.
2. El pliegue restante se reserva exclusivamente para la evaluación.

Este procedimiento se repite hasta que cada pliegue ha sido utilizado una vez como conjunto de prueba. El desempeño final del modelo se obtiene promediando los resultados de todas las iteraciones, lo que proporciona una estimación más estable y robusta.

La estratificación garantiza que, en cada pliegue, se conserve una proporción similar de instancias pertenecientes a cada clase. De este modo, se evitan evaluaciones artificialmente optimistas —asociadas a pliegues con casos sencillos— o excesivamente pesimistas, que no reflejan adecuadamente la distribución real del problema abordado en esta tesis.

Desde una perspectiva intuitiva, este proceso equivale a realizar ocho experimentos independientes, en los cuales el modelo se entrena con diferentes subconjuntos de los datos y se evalúa siempre sobre información no vista. Este enfoque permite medir de forma más confiable la capacidad de generalización del sistema y reduce el riesgo de conclusiones sesgadas.

Búsqueda de hiperparámetros y predicciones OOF

En lugar de fijar los hiperparámetros “a ojo”, se emplea una búsqueda sistemática (*grid search*) combinada con la validación cruzada. Para cada modelo (árbol de decisión o bosque aleatorio) se define una rejilla de valores razonables para sus hiperparámetros principales

(profundidad máxima del árbol, número mínimo de ejemplos por hoja, número de árboles en el bosque, etc.). Cada combinación se evalúa usando como puntaje el área bajo la curva ROC (AUC-ROC).

La configuración que obtiene el mejor promedio de AUC-ROC en los ocho pliegues se selecciona como candidata final. Con ese mejor modelo se calculan las llamadas predicciones *out-of-fold* (OOF): para cada estudio se guarda la predicción emitida por el modelo en la iteración donde ese estudio estuvo en el pliegue de prueba. Así, cada registro se evalúa siempre con un modelo que *no lo vio* durante el entrenamiento.

Estas predicciones OOF permiten estimar de manera honesta métricas como exactitud, sensibilidad o AUC-ROC, sin hacer trampa evaluando sobre datos ya vistos.

Evitar fugas de información (*leakage*)

Un riesgo importante en cualquier proyecto de aprendizaje automático es el *leakage*, es decir, que el modelo tenga acceso, de forma directa o indirecta, a información del conjunto de prueba mientras se está entrenando. Esto puede ocurrir, por ejemplo, si se calculan estadísticos globales (media, desviación estándar, etc.) utilizando *todos* los datos y luego se aplican tanto a entrenamiento como a prueba.

La regla general es:

Toda transformación que “vea” los datos debe ajustarse *solo* con los registros de entrenamiento de cada pliegue, y después aplicarse al pliegue de prueba.

En el caso de los modelos basados en árboles (CART y Random Forest), el reescalado de variables no es imprescindible, lo que simplifica la tubería y reduce las oportunidades de fuga de información. Si en futuros trabajos se incorporan otros clasificadores más sensibles al escalado (por ejemplo, SVM o k -NN), se recomienda utilizar estructuras como `Pipeline` de `scikit-learn`, que se encargan de aplicar las transformaciones en el orden correcto dentro de la validación cruzada.

De esta forma, las métricas reportadas reflejan de manera más fiel el comportamiento esperado cuando el sistema se utilice con pacientes nuevos en un entorno clínico real.

4.3. Balanceo de clases

En numerosos problemas de clasificación supervisada es frecuente que una de las clases esté representada por un número considerablemente mayor de ejemplos que la otra. Esta situación, conocida como *desequilibrio de clases*, puede inducir a evaluaciones engañosas si no se maneja de forma adecuada durante el entrenamiento del modelo.

Cuando existe un fuerte desbalance, un clasificador puede aprender un comportamiento trivial: predecir sistemáticamente la clase mayoritaria y, aun así, obtener valores altos de *accuracy*. Sin embargo, este tipo de solución carece de utilidad práctica, ya que falla precisamente en la identificación de los casos pertenecientes a la clase minoritaria, que suelen ser los de mayor interés en aplicaciones reales.

De manera ilustrativa, si en un conjunto de datos el 90 % de las instancias pertenece a una sola clase, un modelo que siempre prediga dicha clase alcanzará una exactitud del 90 %,

aunque no posea capacidad discriminativa alguna. Este ejemplo pone de manifiesto que la exactitud, por sí sola, no es una métrica suficiente en escenarios con clases desbalanceadas.

Estrategias de balanceo

Para mitigar este problema, se implementa un mecanismo de balanceo configurable que permite controlar cuántos ejemplos de cada clase se utilizan durante el entrenamiento. Desde un punto de vista conceptual, se consideran tres estrategias principales:

- **Entrenamiento sin balanceo.**

Se emplean todos los registros disponibles, respetando la proporción original de las clases. Este escenario sirve como línea base para analizar el comportamiento del modelo cuando no se aplica ninguna corrección al desbalance.

- **Balanceo automático según la clase minoritaria.**

Se selecciona, para cada clase, un número de ejemplos igual al tamaño de la clase con menor representación. Las clases mayoritarias se submuestran y, en caso necesario, las minoritarias pueden remuestrearse con reemplazo, obteniendo así un conjunto de entrenamiento balanceado.

- **Balanceo manual por número fijo.**

Se define explícitamente un número objetivo de ejemplos por clase y se ajusta el conjunto de datos mediante submuestreo o remuestreo hasta alcanzar dicho valor. Esta estrategia permite explorar distintos compromisos entre tamaño del conjunto de entrenamiento y equilibrio entre clases.

En los experimentos documentados en este capítulo se adopta un esquema balanceado, con el mismo número de instancias por clase. Esta decisión permite evaluar de manera más equitativa el desempeño del modelo y evita que la clase mayoritaria domine el proceso de aprendizaje.

En la práctica, el balanceo de clases hace explícito el compromiso (*trade-off*) entre características de desempeño contrapuestas, tales como:

- **sensibilidad**, asociada a la capacidad del modelo para detectar instancias de la clase minoritaria, y
- **especificidad**, relacionada con la correcta identificación de instancias de la clase mayoritaria.

Dependiendo del contexto de aplicación, puede priorizarse un modelo más sensible, aceptando un mayor número de falsas alarmas, o uno más específico, reduciendo dichas alarmas a costa de una menor detección de la clase minoritaria.

4.4. Modelos considerados

Una vez preparado el conjunto de datos, es necesario elegir el tipo de modelo con el que se va a trabajar. En este trabajo se utilizan principalmente dos algoritmos:

1. el **árbol de decisión** tipo CART, y
2. el **bosque aleatorio** (*Random Forest*).

Ambos pertenecen a la familia de modelos basados en árboles, ampliamente estudiados en la literatura de aprendizaje automático. Una ventaja práctica de estos métodos es que permiten inspeccionar su lógica de decisión con mayor facilidad que otros enfoques más opacos, lo cual resulta útil cuando se requiere justificar por qué se produjo una predicción.

Árboles de Clasificación y Regresión (CART)

Los *Árboles de Clasificación y Regresión* (CART, por sus siglas en inglés: *Classification and Regression Trees*) son un tipo de árbol de decisión propuesto por Breiman et al. [41]. Un árbol de decisión puede entenderse como una secuencia de reglas encadenadas. En cada nodo interno se evalúa una condición sobre alguna característica de entrada, por ejemplo:

“¿La característica x_j es mayor que cierto umbral?”

Según la respuesta (sí/no), la instancia avanza por una rama u otra hasta llegar a una hoja, donde se asigna la clase predicha.

Flujo del algoritmo CART. De manera general, el algoritmo construye el árbol de forma recursiva: (i) evalúa posibles particiones del conjunto de datos en un nodo; (ii) elige la partición que más reduce la impureza; (iii) repite el proceso en los nodos hijos; y (iv) detiene el crecimiento cuando se cumple algún criterio (por ejemplo, una profundidad máxima o un mínimo de ejemplos por nodo). La Figura 4.1 resume este procedimiento.

Criterio de partición. Para decidir cómo dividir los datos en cada nodo se utiliza un criterio de *impureza*, como el índice de Gini [41]:

$$G(t) = 1 - \sum_c p_c(t)^2,$$

donde $p_c(t)$ es la proporción de ejemplos de la clase c en el nodo t . Un valor cercano a 0 indica que el nodo es “puro” (predomina una sola clase), mientras que valores más altos indican mezcla de clases. El algoritmo busca particiones que reduzcan la impureza, es decir, que mejoren la separación entre clases.

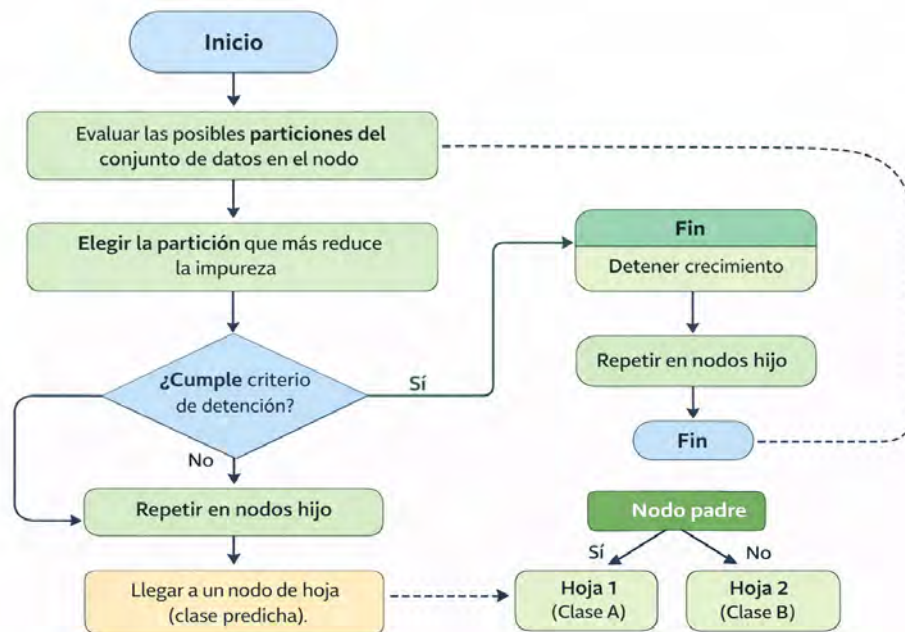


Figura 4.1: Diagrama de flujo del algoritmo CART para construir un árbol de decisión.

Control de complejidad. Si se permite que el árbol crezca sin límite, puede terminar memorizando detalles específicos del conjunto de entrenamiento (*sobreajuste*). Para evitarlo se controlan, entre otros, los siguientes hiperparámetros:

- profundidad máxima del árbol (`max_depth`),
- número mínimo de ejemplos para formar una hoja (`min_samples_leaf`),
- número mínimo de ejemplos para dividir un nodo (`min_samples_split`),
- ponderación de clases (`class_weight`), útil cuando existe desbalance.

En términos prácticos, estos parámetros obligan al árbol a basar sus decisiones en patrones que se repiten con suficiente frecuencia, en lugar de crear ramas altamente específicas para un pequeño número de casos.

Ventaja principal: interpretabilidad. La ventaja del árbol de decisión es que puede expresarse como reglas explícitas. De forma ilustrativa:

Si $x_1 > \tau_1$ y $x_2 \leq \tau_2$, entonces predecir la clase c .

Este tipo de reglas facilita auditar el comportamiento del modelo y relacionar sus decisiones con las variables de entrada utilizadas.

Bosque aleatorio (Random Forest)

Un *Random Forest* (bosque aleatorio) es un ensamble de múltiples árboles de decisión entrenados con aleatoriedad controlada [42]. Cada árbol se entrena con una variante del conjunto de datos, y la predicción final se obtiene por consenso (votación) o mediante el promedio de probabilidades.

Flujo del método. De manera general, el entrenamiento construye N árboles. Para cada árbol se toma una muestra *bootstrap* del conjunto de entrenamiento y, en cada nodo, la mejor partición se busca evaluando únicamente un subconjunto aleatorio de características. En inferencia, cada árbol emite una predicción (o una probabilidad) y el ensamble agrega los resultados para producir la salida final. La Figura 4.2 resume este proceso.

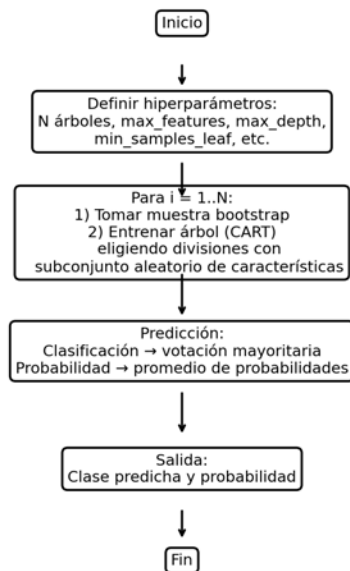


Figura 4.2: Diagrama de flujo del entrenamiento y predicción de un bosque aleatorio (Random Forest).

Pseudocódigo. La lógica del entrenamiento y de la predicción puede resumirse como se muestra en el Pseudocódigo 4.1.

Listing 4.1: Pseudocódigo del entrenamiento y predicción de un Random Forest

```

Entrada: D = {(xi, yi)}, N (n_estimators), m (max_features), criterio, t (umbral
)
# Entrenamiento
Para i = 1..N:
Di <- muestra bootstrap de D (con reemplazo)
Entrenar árbol Ti sobre Di:
Mientras no se cumpla criterio de detencion:

```

```

Seleccionar al azar m características
Elegir la particion que mas reduce la impureza (p.ej., Gini)
Crear nodos hijo y repetir

# Prediccion
Dada una instancia x:
p_i <- probabilidad estimada por cada T_i para la clase 1
p <- (1/N) * sum_i p_i
y_hat <- 1 si p >= t, en otro caso 0 (o votacion mayoritaria)
Salida: y_hat, p

```

Ideas básicas. El entrenamiento incorpora dos fuentes principales de aleatoriedad:

- se generan muestras *bootstrap* del conjunto de entrenamiento (subconjuntos al azar con reemplazo);
- en cada nodo, se evalúa la mejor partición usando solo un subconjunto aleatorio de características.

Esta diversidad induce que distintos árboles cometan errores en regiones diferentes del espacio de características; al agregarlos, el resultado suele ser más estable y con mejor capacidad de generalización que un árbol individual.

Hiperparámetros principales. Entre los parámetros que se ajustan destacan:

- número de árboles del ensamble (`n_estimators`),
- profundidad máxima y tamaño mínimo de las hojas (`max_depth`, `min_samples_leaf`),
- número de características consideradas por división (`max_features`),
- ponderación de clases (`class_weight`).

En general, el bosque aleatorio se considera cuando se busca mayor robustez ante variaciones en los datos o cuando la frontera de decisión se sospecha más compleja que la capturable por un solo árbol. Aunque se pierde la simplicidad de un único árbol, el modelo conserva cierto grado de interpretabilidad mediante medidas como la *importancia de características*.

Otros clasificadores explorados

De forma exploratoria se probaron también modelos como redes neuronales multicapa, máquinas de soporte vectorial (SVM) y clasificadores de vecinos más cercanos (k -NN). Estos enfoques pueden lograr buen desempeño en distintos contextos, pero presentan desventajas frecuentes cuando se requiere un ajuste cuidadoso con conjuntos de datos moderados, así como un análisis más transparente del criterio de decisión:

- su interpretación suele ser menos directa;

- requieren ajuste fino de hiperparámetros y pueden ser más sensibles a la escala de las variables;
- el costo computacional de entrenamiento puede ser mayor.

Por estas razones, se adopta CART como clasificador principal y Random Forest como alternativa más robusta, dejando otros modelos como línea complementaria de exploración.

4.5. Selección de hiperparámetros y umbral de decisión

El comportamiento de un modelo no depende solo de los datos, sino también de los valores concretos de sus hiperparámetros. Para evitar decisiones arbitrarias, se utiliza un procedimiento sistemático.

Búsqueda en malla con AUC-ROC

Se define para cada modelo una rejilla discreta de valores plausibles para los principales hiperparámetros, conforme a rangos y configuraciones reportados en la literatura para aplicaciones similares. Por ejemplo, para un árbol de decisión se exploran distintas profundidades máximas (por ejemplo, entre 4 y 6 niveles), distintos tamaños mínimos de hoja y distintos criterios de impureza (**gini** o **entropy**). Para el bosque aleatorio se prueban varios números de árboles (en torno a 400–500), combinados con diferentes profundidades máximas y formas de seleccionar características.

Cada combinación se evalúa mediante validación cruzada estratificada ($k = 8$). Como métrica principal se utiliza el área bajo la curva ROC (AUC-ROC), por ser independiente de un umbral fijo y resumir la capacidad del modelo para separar normales y anormales en todo el rango de probabilidades.

La combinación que alcanza el mejor promedio de AUC-ROC se selecciona como configuración final.

Elección del umbral de decisión

Los modelos basados en árboles no solo proporcionan una etiqueta (Normal/Anormal), sino también una probabilidad estimada $\hat{p} = \mathbb{P}(y = 1 | X)$ de que el estudio sea anormal. Tradicionalmente se usa un umbral fijo $t = 0,5$, pero en un contexto clínico puede ser conveniente ajustar ese valor.

En esta tesis se consideran dos umbrales de referencia:

- **Umbral de Youden** t_{Youden}^* .

Es el valor de t que maximiza la diferencia $\text{TPR}(t) - \text{FPR}(t)$ (índice J de Youden) [43], donde TPR es la tasa de verdaderos positivos (sensibilidad) y FPR la tasa de falsos positivos. Intenta encontrar un equilibrio global entre detectar anormales y no generar demasiadas falsas alarmas.

- **Umbral de máxima exactitud** t_{ACC}^* .

Es el valor de t que maximiza la proporción total de aciertos (*accuracy*) sobre todas las observaciones.

En términos sencillos, el umbral de Youden busca un punto de equilibrio “clínicamente razonable”, mientras que el umbral de máxima exactitud busca maximizar el porcentaje global de aciertos, aunque eso pueda implicar dejar escapar algunos casos anormales en escenarios muy desequilibrados.

En la implementación actual, el script ajusta el umbral hasta encontrar un compromiso en el que la *accuracy* y el *recall* de la clase anormal sean muy similares. El valor numérico resultante (por ejemplo, $t^* = 0,53$) se guarda como parte de la configuración del modelo.

4.6. Métricas de desempeño

Para valorar qué tan bien distingue el modelo entre estudios *Normales* y *Anormales* se utilizan métricas estándar derivadas de la *matriz de confusión* (*confusion matrix*). En este trabajo se considera como *clase positiva* a la categoría *Anormal* ($y = 1$). La matriz de confusión resume cuántos aciertos y errores comete el modelo:

- **TP (True Positives)**: anormales correctamente etiquetados como *Anormal*.
- **TN (True Negatives)**: normales correctamente etiquetados como *Normal*.
- **FP (False Positives)**: normales mal etiquetados como *Anormal* (falsa alarma).
- **FN (False Negatives)**: anormales mal etiquetados como *Normal* (casos no detectados).

A partir de estos cuatro valores se definen:

$$\text{Exactitud (Accuracy)} = \frac{TP + TN}{TP + TN + FP + FN},$$

$$\text{Precisión (Precision)} = \frac{TP}{TP + FP},$$

$$\text{Sensibilidad (Recall)} = \frac{TP}{TP + FN},$$

$$F_1 = \frac{2 \text{Precisión} \cdot \text{Sensibilidad}}{\text{Precisión} + \text{Sensibilidad}}.$$

En términos sencillos:

- la **exactitud** indica qué porcentaje total de estudios se clasifica correctamente;
- la **precisión** responde: “de todos los estudios que el modelo marcó como anormales, ¿qué porcentaje realmente lo era?”;

- la **sensibilidad** responde: “de todos los estudios anormales, ¿qué porcentaje detectó el modelo?”;
- el **F₁** combina precisión y sensibilidad en un solo número; solo toma valores altos cuando ambas son simultáneamente elevadas.

Adicionalmente se reportan:

- el **AUC-ROC** (área bajo la curva ROC), que mide la capacidad del modelo para separar normales y anormales al variar el **umbral de decisión**;
- la **curva precisión–sensibilidad** (precision–recall), que resulta especialmente informativa cuando la **clase positiva** (*Anormal*) es minoritaria;
- métricas de **calibración** (como el puntaje de Brier), que evalúan si las probabilidades estimadas por el modelo son consistentes con las frecuencias observadas. En clasificación binaria, el *puntaje de Brier* se define como el error cuadrático medio entre la probabilidad predicha y la etiqueta real:

$$\text{Brier} = \frac{1}{n} \sum_{i=1}^n (\hat{p}_i - y_i)^2,$$

donde $y_i \in \{0, 1\}$ y $\hat{p}_i = \mathbb{P}(y_i = 1 \mid X_i)$. Valores más bajos indican mejor calibración (p. ej., predicciones cercanas a 0 o 1 cuando corresponde, y cercanas a la frecuencia real cuando hay incertidumbre). En la implementación actual se reportan AUC-ROC y métricas derivadas de la matriz de confusión; el puntaje de Brier se considera como métrica complementaria para evaluar calibración en trabajo futuro.

En un entorno clínico suele priorizarse una sensibilidad alta para la clase *Anormal* (minimizar falsos negativos y evitar omitir pacientes en riesgo), sin incrementar en exceso las falsas alarmas (falsos positivos), con el fin de no saturar al personal médico.

4.7. Explicabilidad e interpretación

En aplicaciones donde es importante justificar las decisiones del sistema, resulta deseable que el modelo no se comporte como una “caja negra”. Por ello, además del desempeño, se considera la capacidad de ofrecer explicaciones comprensibles a partir de las variables de entrada.

En modelos basados en árboles pueden considerarse, al menos, dos niveles complementarios de explicabilidad:

1. la **importancia de variables**, y
2. la **extracción de reglas**.

Importancia de variables

En modelos basados en árboles, la contribución de cada característica puede estimarse a partir de la reducción de impureza que produce al generar particiones. En un nodo t , una partición elegida por el algoritmo reduce la impureza (por ejemplo, Gini o entropía) en una cantidad $\Delta I(t) = I(t) - (w_L I(t_L) + w_R I(t_R))$, donde t_L y t_R son los nodos hijo y w_L, w_R representan la proporción de muestras que caen en cada hijo. La *importancia por impureza* de una variable x_j se obtiene sumando $\Delta I(t)$ en todos los nodos donde se utilizó x_j para dividir, y normalizando posteriormente el resultado.

En un bosque aleatorio, esta importancia se agrega sobre todos los árboles del ensamble (promedio o suma acumulada) y suele reportarse como un valor no negativo por variable. Tras la normalización, los valores suelen quedar en el intervalo $[0, 1]$ y su suma es 1, por lo que pueden interpretarse como una *proporción relativa* de contribución: a mayor valor, mayor impacto promedio de esa variable en las decisiones del modelo. En consecuencia, se obtiene un ranking de variables desde las más influyentes hasta las menos relevantes.

Cabe señalar que esta medida es *relativa* al conjunto de datos y al modelo entrenado: no es un “porcentaje de causalidad” ni implica que una sola variable determine el diagnóstico. Además, puede sesgarse a favor de variables continuas o con muchos valores posibles, por lo que, cuando se requiere un análisis más robusto, puede complementarse con métodos como la importancia por permutación o técnicas de explicabilidad locales.

Extracción de reglas

En un árbol de decisión, cada trayectoria desde la raíz hasta una hoja puede expresarse como una regla SI-ENTONCES que combina umbrales sobre variables de entrada. En otras palabras, cada nodo interno aporta una condición (p. ej., $x_j \leq \tau$) y la regla final se construye como la conjunción (lógica AND) de todas las condiciones que se recorren hasta llegar a una hoja. La clase predicha corresponde a la etiqueta asociada a dicha hoja, y la *confianza* puede interpretarse a partir de la distribución de clases en ese nodo (por ejemplo, la proporción de anormales y normales).

La Figura 4.3 muestra un ejemplo de árbol de decisión. Para extraer una regla a partir del árbol, se siguen los pasos:

1. Iniciar en la raíz y registrar la condición mostrada en el nodo.
2. Elegir la rama según el resultado de la condición (True/False) y repetir el proceso en el siguiente nodo.
3. Al llegar a una hoja, formar la regla SI-ENTONCES con todas las condiciones registradas; el consecuente de la regla es la clase asignada en la hoja.

Por ejemplo, si se sigue en la Figura 4.3 la ruta: (i) $pr_interval \leq 0,071$ y luego (ii) $qrs_duration \leq 0,095$, y posteriormente (iii) $st_qt_ratio \leq 0,794$, se obtiene una regla de la forma:

SI $pr_interval \leq 0,071$ *Y* $qrs_duration \leq 0,095$ *Y* $st_qt_ratio \leq 0,794$, *ENTONCES* predecir **Normal**.

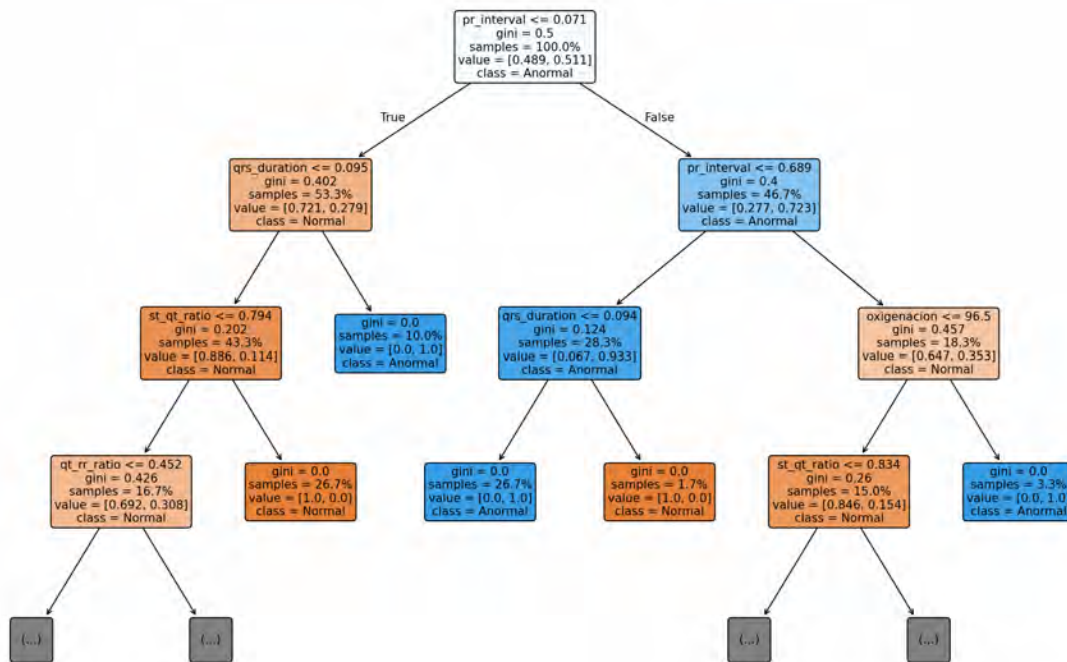


Figura 4.3: Ejemplo de árbol de decisión para clasificación. Cada ruta raíz–hoja define una regla SI–ENTONCES mediante la concatenación de las condiciones de los nodos.

Este mecanismo hace que el modelo sea explícito y revisable: cada predicción puede justificarse indicando la ruta seguida y los umbrales aplicados. Además, campos comunes en la visualización del árbol (por ejemplo, *gini*, *samples* y *value*) ayudan a interpretar la *pureza* del nodo, la cantidad de ejemplos que soportan esa regla y la distribución de clases en la hoja.

En modelos de ensamble como Random Forest, donde participan múltiples árboles, la interpretación puede apoyarse en medidas agregadas de importancia de variables y, opcionalmente, en la selección de un árbol representativo o en la obtención de reglas frecuentes (aquellas que aparecen repetidamente en el ensamble). Estas herramientas no sustituyen el criterio del usuario experto, pero permiten auditar el modelo y discutir sus decisiones de forma más transparente.

4.8. Persistencia del modelo y paquete de despliegue

Una vez entrenado un modelo, es necesario conservarlo de forma reproducible para su uso en predicción sobre nuevos datos. En la práctica, no basta con almacenar únicamente el estimador (entrenado), ya que también se requiere preservar la configuración exacta con la que fue construido.

Para ello, es común definir un *paquete de despliegue* que contenga, de manera estructurada:

- el modelo entrenado;

- la lista y el orden de las características de entrada esperadas;
- los hiperparámetros seleccionados y la configuración de validación;
- la política de preprocesamiento aplicada (por ejemplo, normalización o recortes);
- el criterio de decisión utilizado (por ejemplo, umbral sobre probabilidad, si aplica);
- metadatos de versión (fecha de entrenamiento, conjunto de entrenamiento, etc.).

Este enfoque reduce errores por inconsistencias entre entrenamiento y predicción (por ejemplo, cambios de columnas o reordenamientos) y facilita la trazabilidad: permite identificar con qué configuración se generó cada predicción y reproducir el modelo cuando sea necesario.

4.9. Evaluación del clasificador: descripción del protocolo

La evaluación experimental del clasificador —incluyendo la generación de predicciones fuera de muestra (*out-of-fold*, OOF), la selección del umbral de decisión, el análisis mediante matriz de confusión y la obtención de curvas ROC y precisión–recall— se presenta de forma íntegra en el Capítulo 6, particularmente en las Secciones 6.1 y 6.2. En este capítulo únicamente se describe el flujo experimental y la configuración utilizada para asegurar la reproducibilidad.

4.10. Buenas prácticas y reproducibilidad

Para que el sistema pueda evolucionar de forma ordenada (por ejemplo, al incorporar más pacientes o nuevas características), se han adoptado varias buenas prácticas de ingeniería de datos y de software:

- **Semillas fijas de aleatoriedad.**
Se establece un valor fijo `RNG_SEED` para todos los procesos que implican aleatoriedad (particiones, remuestreos, entrenamiento). Esto permite que una misma configuración produzca siempre el mismo modelo y las mismas métricas.
- **Registro de experimentos.**
Cada corrida guarda la fecha, el tipo de modelo, la rejilla de hiperparámetros explorada, la combinación ganadora, las métricas OOF (ACC, Recall, AUC, etc.), el umbral t^* y la lista de características utilizadas. Esto facilita comparar corridas, detectar errores y justificar decisiones.
- **Versionado del código y de los modelos.**
Cada archivo de modelo se asocia a una versión concreta del código de entrenamiento. De este modo, si el desempeño cambia en una versión posterior, se puede rastrear el origen de la diferencia.

- **Almacenamiento de gráficas.**

Se guardan las curvas ROC y PR, las matrices de confusión y las gráficas de importancia de variables. Estas figuras son útiles tanto para la evaluación técnica como para la comunicación con el equipo clínico.

- **Revisión clínica periódica.**

Los falsos positivos y falsos negativos se revisan periódicamente con el cardiólogo, especialmente aquellos procedentes del Hospital Central. Este ciclo de retroalimentación permite corregir posibles errores de etiquetado y detectar patrones que justifiquen nuevos ajustes del modelo.

Estas prácticas buscan que el entrenamiento del modelo deje de ser un proceso artesanal y se convierta en un procedimiento documentado, repetible y auditable.

4.11. Resumen del capítulo

En este capítulo se describió el flujo completo para entrenar y evaluar modelos de aprendizaje automático aplicados a las características temporales del ECG. Se explicó cómo se divide el conjunto de datos en pliegues de validación cruzada estratificada, cómo se equilibran las clases para evitar que el modelo se limite a predecir siempre “Normal” y cómo se realiza una búsqueda sistemática de hiperparámetros basada en AUC-ROC.

Se justificó la elección de los árboles de decisión (CART) como modelo principal por su interpretabilidad, y de los bosques aleatorios como alternativa más robusta frente al ruido. También se presentaron las principales métricas de clasificación (accuracy, precisión, sensibilidad, F_1 , AUC-ROC) y se discutió la importancia de ajustar el umbral de decisión para equilibrar la detección de casos anormales y el número de falsas alarmas.

Se mostraron resultados experimentales reales sobre un subconjunto balanceado de 90 estudios (45 normales y 45 anormales), en los que el modelo Random Forest alcanza un desempeño alrededor del 81 % de exactitud OOF y un AUC-ROC de 0.872. Finalmente, se describió el esquema de persistencia del modelo mediante un paquete de despliegue y las buenas prácticas adoptadas para garantizar la reproducibilidad y trazabilidad de los experimentos.

En el capítulo siguiente se detallará cómo este modelo se integra en el sistema completo de diagnóstico cardiovascular asistido, desde la captura de la señal ECG hasta la presentación de resultados en la plataforma web.

Capítulo 5

Desarrollo del sistema de diagnóstico cardiovascular asistido

En los capítulos anteriores se describió, por separado, cómo se adquieren las señales de ECG con *OpenBCI Ganglion*, cómo se procesan digitalmente y cómo se entrena un modelo de aprendizaje automático capaz de distinguir entre registros *Normales* y *Anormales*.

Este capítulo reúne todas esas piezas y las presenta como un sistema completo, pensado para uso real: desde que el paciente se coloca los electrodos hasta que el cardiólogo visualiza la traza en la pantalla y recibe una sugerencia automática de diagnóstico. Se explica cómo se combinan **OpenBCI Ganglion**, **Python + BrainFlow** y una capa web en **PHP + MySQL + Plotly** para:

- capturar el ECG en tiempo real,
- procesarlo y extraer características relevantes,
- guardar la información con trazabilidad,
- aplicar el modelo de clasificación,
- y presentar los resultados de forma clara al personal clínico.

Además de la arquitectura técnica, se aborda la vinculación con el Hospital Central de San Luis Potosí para validar el sistema, el rendimiento observado en pruebas reales y el plan de pruebas utilizado, así como la confidencialidad de la información.

5.1. Requerimientos del Sistema

El sistema de diagnóstico cardiovascular asistido se concibió con un objetivo central: ofrecer una herramienta de bajo costo y alto valor práctico que acompañe al cardiólogo en la interpretación de electrocardiogramas, sin sustituir en ningún momento su criterio clínico. La visión es que un médico o un residente, aun sin conocimientos profundos de procesamiento de señales o aprendizaje automático, pueda:

1. realizar un estudio ECG de forma rápida con un dispositivo portátil,

2. ver la traza con una cuadrícula similar a la de un electrocardiograma convencional,
3. consultar de inmediato las mediciones clave (RR, PR, QRS, QT, QTc, BPM, etc.),
4. y recibir una sugerencia automática de si el patrón es compatible con un registro normal o si amerita una revisión más detallada.

Para lograrlo, el sistema se organiza en torno a seis funciones principales:

- **Capturar** ECG en tiempo real con **OpenBCI Ganglion** (200 Hz) desde Python usando **BrainFlow**. En la práctica, esto se traduce en un programa que el operador ejecuta en la computadora: al iniciar la captura, el sistema se conecta vía Bluetooth a la placa, muestra en pantalla la traza en tiempo real y registra 30 segundos de señal filtrada por cada estudio.
- **Filtrar** la señal en cero fase (detrending, filtro pasa-banda 0.5–45 Hz y *notch* a 60 Hz) para preservar la morfología P–QRS–T. De esta manera, la señal que llega al resto del sistema ya está libre, en gran medida, de deriva de línea de base, zumbido de red y ruido muscular, manteniendo las ondas con la forma que el médico espera ver.
- **Detectar** ondas y puntos característicos (P, complejo QRS, T; y puntos Q y S alrededor de R) y **calcular** los intervalos fundamentales (RR, PR, QRS, QT, ST) y magnitudes derivadas (QTc, frecuencia cardíaca en BPM, duración de la onda P). Estos valores se calculan en Python inmediatamente después de la captura y representan el “resumen numérico” del ECG.
- **Persistir** la información en formatos adecuados:
 - un archivo CSV con la señal filtrada (30 s a 200 Hz, una columna por derivación),
 - una imagen PNG con cuadrícula clínica (1 mV vertical, 200 ms horizontal) para revisión visual,
 - y el registro de metadatos en una base de datos MySQL/MariaDB, incluyendo un folio único, edad, sexo, fecha/hora, valores de intervalos y, cuando está disponible, diagnóstico médico.

Para asegurar la trazabilidad, puede registrarse también un hash criptográfico (por ejemplo, SHA-256) de los archivos CSV y PNG.

- **Entrenar** modelos interpretables (árboles de decisión CART y, según se requiera, bosques aleatorios) y **predecir** la condición *Normal/Anormal* con un umbral óptimo (índice de Youden o maximización de la exactitud). El modelo se entrena de forma *offline* con la base de datos acumulada y, una vez consolidado, se despliega como un archivo `modelo_ecg.pkl` que puede ser cargado por los scripts de Python en el momento de la captura o por la capa web cuando se consulta un estudio.

- **Visualizar** la traza y las características en una interfaz web desarrollada en **PHP + MySQL + Plotly**, habilitando una revisión clínica rápida. Desde esta interfaz (accesible únicamente para usuarios autorizados) el médico puede seleccionar un **folio**, ver la señal con cuadrícula, revisar los intervalos calculados, consultar la sugerencia del modelo (Normal/Anormal) y, en caso necesario, registrar su propio diagnóstico definitivo.

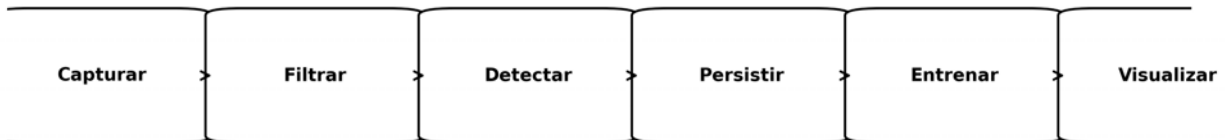


Figura 5.1: Diagrama de bloques de las funciones principales del sistema.

La validación del sistema no se limita a pruebas sintéticas o bases de datos públicas, sino que se apoya en **casos anormales reales capturados y validados en el Hospital Central de San Luis Potosí**. Estos casos se han utilizado como *ground truth* en el entrenamiento supervisado y en la revisión de falsos positivos y falsos negativos, permitiendo ajustar tanto la lógica de detección de ondas como el umbral de decisión del modelo.

Con esta visión, el sistema pretende ser un puente entre la ingeniería y la práctica clínica cotidiana: por un lado aprovecha técnicas modernas de procesamiento de señales y aprendizaje automático; por otro, se esfuerza por presentarlas en una forma que resulte útil, comprensible y confiable para el equipo de salud que atiende directamente a los pacientes.

Antes de construir cualquier sistema orientado a la práctica clínica es necesario aclarar qué se espera que haga (requerimientos funcionales) y bajo qué condiciones debe hacerlo (requerimientos no funcionales). En este proyecto se parte de una idea sencilla: el sistema debe acompañar al médico desde la captura del ECG hasta la visualización del resultado asistido, sin exigirle conocimientos avanzados de programación o de procesamiento de señales.

Requerimientos funcionales

Los requerimientos funcionales definen las acciones concretas que el sistema debe ser capaz de realizar. En este caso, se identificaron los siguientes:

1. **Conectar a la placa Ganglion vía BLE y mostrar la señal en tiempo real.** El programa en Python debe establecer la conexión Bluetooth Low Energy con *OpenBCI Ganglion*, iniciar el flujo de datos (*stream*) y mostrar al operador la señal en tiempo real, de modo que pueda verificar que los electrodos están bien colocados y que la traza tiene una morfología razonable antes de guardar el estudio.
2. **Acondicionamiento de las señales ECG** El sistema debe aplicar en tiempo real el esquema de preprocesamiento definido (detrending, filtro pasa-banda 0.5–45 Hz y filtro

notch a 60 Hz) y ejecutar el algoritmo de detección de ondas y puntos característicos (P, QRS, T; y Q/S alrededor de R). Esto permite que, al término del registro, ya se disponga de una primera identificación automática de los latidos y de los intervalos.

3. **Calcular y enviar características y archivos al servidor.** Finalizada la captura, el programa debe calcular las características principales (RR, PR, QRS, QT, ST, QTc, BPM, duración de P, etc.), guardar los archivos CSV y PNG correspondientes y enviarlos, junto con los metadatos del paciente (edad, sexo, fecha/hora, folio), a un servidor web mediante una petición HTTP.
4. **Visualizar registros, curvas y características en la interfaz web.** La aplicación web debe permitir que un usuario autenticado seleccione un estudio y visualice: la traza ECG con cuadrícula clínica (vía Plotly), las características calculadas y, si ya existe, el diagnóstico automático del modelo y el diagnóstico definitivo del médico. Todo ello en una pantalla clara y fácil de interpretar.
5. **Entrenar modelos de Aprendizaje Automático.** El sistema debe contar con un módulo de entrenamiento que, a partir de los registros almacenados en la base de datos, descargue las características relevantes, ejecute la validación cruzada con búsqueda en malla (Cap. 4) y genere un *paquete de despliegue (bundle)* que concentre el modelo entrenado, sus hiperparámetros y el umbral de decisión. En la implementación, este *bundle* se materializa como un archivo serializado (por ejemplo, `modelo_ecg.pkl`) y un archivo de configuración asociado (por ejemplo, `config.json`) para asegurar que la predicción se realice con la misma configuración usada durante el entrenamiento.
6. **Predecir en lote registros pendientes y registrar resultados.** Periódicamente, un servicio en Python debe revisar qué registros aún no cuentan con diagnóstico automático, aplicar el modelo desplegado, registrar la predicción (Normal/Anormal), la probabilidad asociada y las características internas, y dejar constancia de ello en la base de datos para su posterior revisión clínica.

Requerimientos no funcionales

Los requerimientos no funcionales describen las cualidades que debe tener el sistema para ser realmente útil en un entorno clínico: facilidad de uso, tiempo de respuesta razonable, fiabilidad, trazabilidad y cumplimiento de aspectos relacionados con el acceso y la protección de datos.

- **Facilidad de operación.** La interfaz de usuario, tanto del programa de captura en PC como de la aplicación web, debe ser simple e intuitiva. El personal de salud no tiene por qué conocer detalles técnicos del hardware o del modelo; se busca que las acciones se asimilen a las de un electrocardiógrafo convencional (iniciar estudio, detener, visualizar, imprimir o consultar), con el valor añadido de las características automáticas y la visualización digital.
- **Rendimiento.** La latencia de captura debe ser menor o igual a 150 ms, de manera que la señal en pantalla se perciba prácticamente en tiempo real, permitiendo corregir a

tiempo el montaje si es necesario. La inferencia del modelo (cálculo de la probabilidad de anormalidad y decisión Normal/Anormal) debe tomar menos de 5 ms por registro, para que el resultado esté disponible de forma inmediata al consultar un estudio.

- **Fiabilidad.** El sistema debe ser resistente a fallos de comunicación. Esto incluye mecanismos de reconexión BLE (si se pierde momentáneamente el enlace con la placa Ganglion), reintentos en las peticiones HTTP hacia el servidor y validación estricta de los *payloads* recibidos (por ejemplo, verificar que las columnas de los CSV sean las esperadas y que las características tengan valores plausibles).
- **Autenticación de accesos.** El acceso a la interfaz web debe estar controlado mediante mecanismos de autenticación de usuarios y control de roles, permitiendo diferenciar, por ejemplo, entre perfiles de personal médico y personal técnico. Este control garantiza que cada usuario pueda acceder únicamente a las funciones y a la información que le corresponden dentro del sistema.
- **Protección de datos y aviso de privacidad.** El sistema debe operar conforme a un aviso de privacidad que informe sobre el tratamiento de los datos clínicos y personales de los pacientes. La comunicación con el servidor se realiza mediante HTTPS para proteger la información en tránsito, y los archivos almacenados (CSV y PNG) pueden asociarse a mecanismos de verificación de integridad, de modo que cualquier modificación no autorizada pueda ser detectada. El manejo de los datos se orienta a preservar la confidencialidad y el uso responsable de la información clínica.
- **Trazabilidad.** Cada estudio debe identificarse mediante un folio único que permita seguirlo a lo largo de todo el sistema: desde la captura local hasta la visualización web y la revisión posterior. Adicionalmente, se mantienen bitácoras de eventos relevantes que facilitan la auditoría y el seguimiento de los estudios realizados.

En conjunto, estos requerimientos buscan que el sistema no solo funcione en un entorno de laboratorio, sino que pueda integrarse, con las adaptaciones necesarias, a un entorno clínico u hospitalario real.

5.2. Arquitectura general del sistema

Desde un punto de vista de arquitectura, el sistema se organiza en dos grandes bloques que se comunican entre sí: una **capa de adquisición local** y una **capa web en servidor**. Esta separación permite que el procesamiento más cercano a la señal (filtrado, detección de ondas, cálculo de intervalos) se realice en el equipo donde está conectada la placa Ganglion, mientras que el almacenamiento a largo plazo, la visualización y el entrenamiento de modelos se realizan en un servidor central.

El sistema adopta, por tanto, una **arquitectura de dos capas** (Figura 5.2) que distingue claramente entre:

1. **Capa de adquisición local** (Python + BrainFlow). En esta capa se ejecuta el programa de captura sobre un ordenador portátil o de escritorio. Sus responsabilidades principales son:

- conectarse a *OpenBCI Ganglion* vía BLE,
 - capturar la señal a 200 Hz,
 - aplicar el filtrado y la estandarización,
 - detectar ondas y puntos característicos,
 - calcular las características cardiológicas (RR, PR, QRS, QT, ST, QTc, BPM, etc.),
 - mostrar la señal en tiempo real mediante PyQtGraph,
 - y, al finalizar el registro, generar y almacenar los archivos CSV/PNG y enviar los datos al servidor.
2. **Capa web** (PHP + MySQL + Plotly). Esta capa reside en un servidor (por ejemplo, en un hosting compartido o un servidor institucional) y se encarga de:
- recibir los archivos y características enviados desde la capa local,
 - guardar la información en una base de datos MySQL/MariaDB, asociando cada estudio a un folio y a un paciente,
 - ofrecer una interfaz web para la visualización de la traza (gráficas interactivas con Plotly) y de las características,
 - exponer puntos de entrada (*endpoints*) para que el módulo de entrenamiento/pre-dicción pueda descargar datos y subir resultados del modelo.

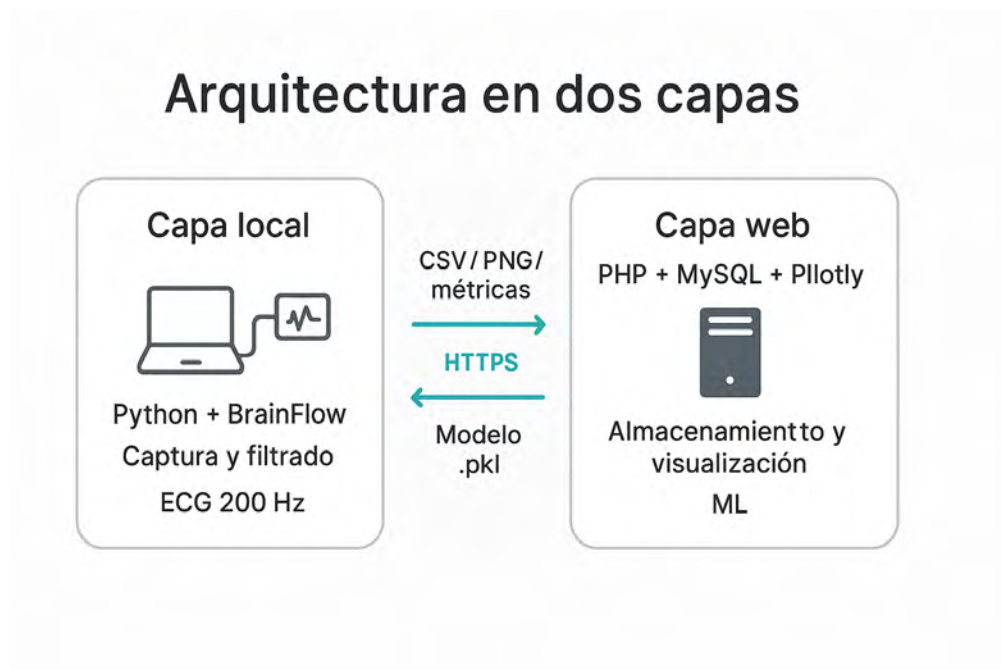


Figura 5.2: Arquitectura de dos capas: adquisición y preprocesamiento local (Python + BrainFlow) y, en el servidor, almacenamiento, visualización web y aprendizaje automático.

Esta separación tiene varias ventajas prácticas: permite usar la misma aplicación web desde distintos consultorios, facilita el respaldo centralizado de la base de datos y la actualización del modelo sin necesidad de modificar el programa de captura en cada equipo.

Flujo de datos de punta a punta

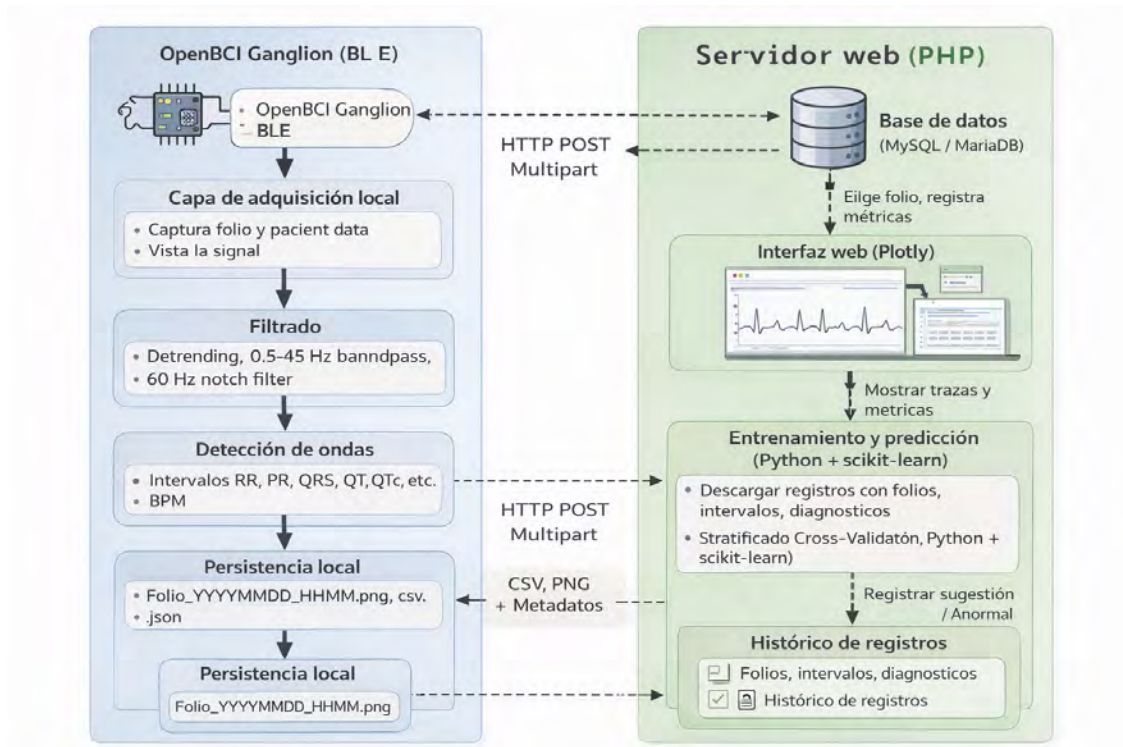


Figura 5.3: Diagrama de flujo de punta a punta del sistema: desde la adquisición de la señal ECG con la placa OpenBCI Ganglion, pasando por el preprocesamiento, extracción de características y persistencia local, hasta el envío al servidor, la visualización web y el entrenamiento y predicción del modelo de aprendizaje automático.

La Figura 5.3 resume de forma visual el recorrido completo que sigue cada estudio, permitiendo identificar claramente qué procesos se ejecutan de manera local y cuáles dependen del servidor central.

Para comprender mejor cómo interactúan ambas capas, resulta útil seguir el flujo de datos de punta a punta, desde la captura de la señal hasta la visualización y la clasificación automática:

1. **Captura.** El operador ejecuta el programa en Python, introduce los datos básicos del paciente y presiona el botón de inicio. El sistema inicializa el objeto `BoardShim` de `BrainFlow`, establece la conexión con la placa Ganglion y comienza el *stream* de datos. En pantalla se observa la señal en tiempo real, lo que permite verificar que el montaje de electrodos es correcto.
2. **Filtrado.** Conforme llegan las muestras, el sistema aplica el preprocesamiento en cero fase: detrending para eliminar la componente de continua, filtro pasa-banda 0.5–45 Hz para concentrarse en la banda cardiológica de interés y filtro *notch* a 60 Hz para suprimir el zumbido de la red eléctrica. El resultado es una señal más limpia, en la que se distinguen claramente las ondas P, QRS y T.

3. **Detección de ondas y puntos.** Sobre la señal filtrada se ejecuta el algoritmo de detección: primero se localizan los picos R mediante un umbral adaptativo y una distancia refractaria mínima entre picos; posteriormente, en ventanas relativas a cada R se buscan las ondas P y T y los puntos Q y S. A partir de las marcas temporales halladas se calculan latencias y amplitudes por latido.
4. **Cálculo de características.** Una vez detectados los hitos, el programa calcula los promedios de los intervalos RR, PR, QRS, QT y ST, la frecuencia cardíaca (BPM), el QT corregido (QTc) y, cuando aplica, la duración de la onda P. También se registran conteos de ondas y se realiza una primera clasificación simple de ritmo y anomalías.
5. **Persistencia local.** Al finalizar la captura, el sistema guarda un CSV con la señal cruda, un CSV con la señal filtrada y una imagen PNG de la traza con malla clínica, simulando el papel milimetrado de un electrocardiógrafo. Los archivos se almacenan localmente con nombres que incorporan el folio y la fecha y hora.
6. **Envío al servidor.** El programa envía los archivos y las características al servidor mediante peticiones HTTP. El servidor recibe los datos, los valida y registra en la base de datos las rutas a los archivos y los valores numéricos asociados al estudio.
7. **Entrenamiento y predicción.** Cuando se desea entrenar un nuevo modelo, un script de Python descarga el histórico de registros etiquetados, ejecuta el proceso de aprendizaje automático y guarda el modelo resultante. Posteriormente, el sistema puede detectar registros pendientes, aplicar el modelo y almacenar en la base de datos la sugerencia de condición clínica asociada a cada folio.

Este flujo de datos asegura que cada estudio atraviese las mismas etapas de procesamiento y evaluación, ofreciendo al médico una experiencia coherente y reproducible basada en un preprocesamiento y un modelo controlados.

5.3. Módulo de captura en tiempo real (Python + BrainFlow)

El módulo de captura en tiempo real es el “primer eslabón” de todo el sistema: aquí es donde el paciente está conectado a la placa, el operador observa la traza en pantalla y se decide si el registro es suficientemente limpio como para guardarlo. Todo lo que suceda en esta etapa impacta de manera directa en la calidad de las características y, por tanto, en la confiabilidad del modelo de clasificación.

Este módulo se desarrolló en **Python** utilizando la biblioteca **BrainFlow**, que ofrece una interfaz de alto nivel para comunicarse con *OpenBCI Ganglion*, iniciar el flujo de datos, aplicar filtros y acceder a las muestras como arreglos numéricos listos para su procesamiento.

Conexión a Ganglion y colocación de electrodos

Para obtener una **derivación II equivalente** con la placa Ganglion en configuración de tres electrodos (*3-lead*), se utilizan los terminales +1, -1 y D_G (BIAS o referencia). Los

cuatro interruptores del tablero se colocan en posición UP, lo que configura las entradas como referidas a D_G, de acuerdo con la recomendación del fabricante.

Siguiendo la documentación oficial de OpenBCI para ECG con tres electrodos (véase la Figura 5.4), la colocación típica es la siguiente:

- **RA (Right Arm):** debajo de la clavícula derecha.
- **LA (Left Arm):** debajo de la clavícula izquierda.
- **LL (Left Leg):** región infracostal izquierda (borde inferior de las costillas), electrodo que actúa como referencia y se conecta al terminal D_G.

En esta configuración, el canal formado por los terminales +1 y -1 mide la diferencia de potencial entre los electrodos RA y LA, mientras que el electrodo LL se utiliza como referencia común (BIAS). El resultado es una señal muy similar a la **derivación II convencional** (RA→LL), ampliamente utilizada para monitorización y diagnóstico cardiológico básico.

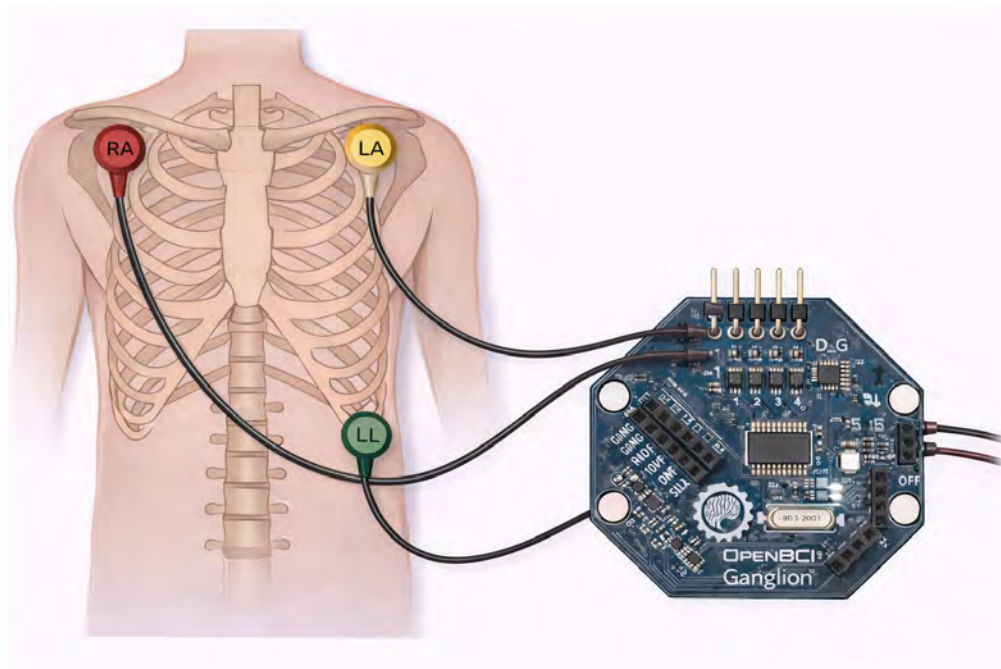


Figura 5.4: Colocación de electrodos y conexión a la placa OpenBCI Ganglion para la obtención de una derivación II equivalente utilizando tres electrodos.

En términos prácticos, el procedimiento que sigue el operador es el siguiente: primero limpia los puntos de contacto con gasas y alcohol isopropílico, coloca los electrodos adhesivos en las posiciones señaladas y conecta los cables a los terminales +1, -1 y D_G de la placa Ganglion. Una vez hecho esto, se inicia el programa de captura; si la traza se observa estable y con la morfología esperada (ondas P visibles y complejos QRS bien definidos), se procede a registrar el estudio.

Secuencia mínima con BrainFlow. BrainFlow facilita la conexión con la placa mediante el objeto `BoardShim`. A continuación se muestra un ejemplo simplificado de la secuencia utilizada en el módulo de captura:

```
from brainflow.board_shim import BoardShim, BrainFlowInputParams, BoardIds

# 1) Definir parámetros de conexión (MAC address del adaptador BLE)
params = BrainFlowInputParams()
params.mac_address = "e6:d4:92:5f:0a:b1" # ejemplo, se ajusta según el dispositivo

# 2) Crear objeto BoardShim para la placa Ganglion
board = BoardShim(BoardIds.GANGLION.value, params)

# 3) Preparar sesión e iniciar el flujo de datos
board.prepare_session()
board.start_stream()

# ... lectura del buffer, filtrado y detección de ondas ...

# 4) Detener el flujo y liberar recursos
board.stop_stream()
board.release_session()
```

En la implementación completa, esta secuencia se complementa con manejo de errores (por ejemplo, intentos de reconexión si falla el enlace BLE) y con la lógica de actualización periódica del gráfico de señal en la interfaz gráfica.

Visualización en tiempo real y control de calidad

Para la visualización en tiempo real se utiliza la biblioteca **PyQtGraph**, que permite graficar señales en tiempo real de forma eficiente. Se emplea un *buffer deslizante* de aproximadamente 4 s de duración, de modo que en la ventana siempre se observa una porción reciente de la traza.

La interfaz muestra:

- un título dinámico con el nombre del proyecto y la institución (por ejemplo, “Facultad de Ingeniería – UASLP – Módulo ECG”),
- un contador de tiempo que indica cuántos segundos de señal se han capturado,
- y, opcionalmente, indicadores visuales de calidad de señal (por ejemplo, un icono o texto que advierte si la amplitud es demasiado baja o si hay saturación).

Además, el módulo calcula de manera continua características rápidas de calidad, como una estimación de la relación señal–ruido (SNR) y el conteo de picos R por minuto. Si estos valores salen de rangos razonables (por ejemplo, un número de R irrealmente alto o bajo para un adulto en reposo), el programa muestra una advertencia al operador para que revise los electrodos o solicite al paciente que permanezca más quieto. Este *feedback* inmediato evita conservar registros que, desde el inicio, se sabe que serán poco útiles para el análisis clínico.

Filtrado y estandarización

Sea $x[n]$ la señal ECG cruda muestreada a $f_s = 200$ Hz. El preprocesamiento se aplica en **fase cero** mediante filtrado hacia adelante y hacia atrás (*forward-backward filtering*), lo que evita desfases temporales (*phase lag*) y preserva la alineación de ondas P, complejo QRS y onda T, indispensable para estimar intervalos con precisión en señales ECG segmentadas *offline* [44]. En forma general, para un filtro IIR definido por coeficientes $\{a_k\}$ y $\{b_k\}$, el filtrado de fase cero puede representarse como:

$$x_f[n] = \text{FBF}\{x[n]\} \equiv \mathcal{F}^{-1}(\mathcal{F}(\mathcal{F}(x[n]))) ,$$

donde $\mathcal{F}(\cdot)$ denota el filtrado causal (hacia adelante) y $\mathcal{F}^{-1}(\cdot)$ el filtrado sobre la señal invertida en el tiempo (hacia atrás), resultando en una respuesta de magnitud al cuadrado y fase aproximadamente nula [44].

El *pipeline* aplicado es:

- **Corrección de línea de base (detrending / *baseline wander*)**. La deriva lenta del ECG (respiración, movimiento, cambios de impedancia electrodo–piel) se concentra típicamente en muy bajas frecuencias y puede sesgar la detección de picos y la medición de segmentos [45]. Se estima una tendencia $\hat{b}[n]$ y se corrige:

$$x_d[n] = x[n] - \hat{b}[n].$$

En esta tesis, $\hat{b}[n]$ puede modelarse mediante:

$$\hat{b}[n] = \sum_{k=0}^p c_k n^k \quad (\text{ajuste polinomial de bajo orden}),$$

o, de forma equivalente, por un suavizado de muy baja frecuencia (promedio móvil), según la rutina empleada en el entorno de ejecución.

- **Filtro pasa–banda (0.5–45 Hz)**. Para conservar la morfología P–QRS–T y atenuar componentes fuera de banda (deriva residual y ruido muscular/EMG), se emplea un pasa–banda cuya banda útil captura la información dominante del ECG con $f_s = 200$ Hz (Nyquist = 100 Hz), evitando además amplificar ruido de alta frecuencia [45, 46]. El filtro se implementa como un IIR (por ejemplo, Butterworth), descrito por su función de transferencia:

$$H_{bp}(z) = \frac{\sum_{k=0}^M b_k z^{-k}}{1 + \sum_{k=1}^N a_k z^{-k}}, \quad x_{bp}[n] = \text{FBF}\{H_{bp}(z), x_d[n]\}.$$

La elección 0.5–45 Hz es un compromiso práctico para análisis de ritmo e intervalos: suprime deriva lenta y reduce EMG, preservando la energía del QRS y la onda T relevantes para mediciones temporales [45, 46].

- **Filtro *notch* a 60 Hz (interferencia de red)**. Para mitigar la componente sinusoidal de la red eléctrica sin distorsionar en exceso la morfología, se aplica un rechazo de banda

angosto centrado en $f_0 = 60$ Hz [45]. Una forma estándar de notch de segundo orden es:

$$H_{\text{notch}}(z) = \frac{1 - 2 \cos(\omega_0)z^{-1} + z^{-2}}{1 - 2r \cos(\omega_0)z^{-1} + r^2 z^{-2}}, \quad \omega_0 = 2\pi \frac{f_0}{f_s},$$

donde $r \in (0, 1)$ controla el ancho de banda (mayor r implica notch más estrecho). El resultado se aplica también en fase cero:

$$x_{nf}[n] = \text{FBF}\{H_{\text{notch}}(z), x_{bp}[n]\}.$$

- **Estandarización z -score.** Finalmente, para hacer comparables amplitudes entre sujetos/sesiones y estabilizar umbrales (p.ej., en detección de picos), se normaliza cada segmento filtrado:

$$z[n] = \frac{x_{nf}[n] - \mu}{\sigma},$$

donde μ y σ son la media y desviación estándar del segmento. Esta etapa reduce la sensibilidad a variaciones de ganancia y contacto de electrodos.

En la implementación se emplean rutinas equivalentes de filtrado IIR y filtrado adelante–atrás disponibles en `DataFilter` de BrainFlow y, cuando aplica, en `scipy`. El objetivo final es conservar la morfología clínica P–QRS–T con la menor cantidad posible de ruido superpuesto, manteniendo la alineación temporal necesaria para el cálculo confiable de intervalos [45, 46].

Detección de P, Q, R, S y T

Sobre la señal ya filtrada y estandarizada se aplica el algoritmo de detección de ondas y puntos característicos. En términos clínicos, este procedimiento busca emular la lectura del cardiólogo: identificar la morfología P–QRS–T, ubicar los puntos (o límites) relevantes y, a partir de ellos, medir intervalos sobre el papel milimétrico. La Figura 5.5 muestra un ejemplo de un latido anotado con los puntos y los intervalos más usados en la práctica clínica [47].

El procedimiento, inspirado en Pan–Tompkins y adaptado para trabajar a 200 Hz, se resume en:

- **Picos R (QRS):** se calcula la derivada de la señal, se eleva al cuadrado y se aplica una integración móvil para resaltar la energía de los complejos QRS. Luego se fija un umbral dinámico y una **distancia mínima de 300 ms** entre picos consecutivos, evitando confundir artefactos o ruido con latidos reales.
- **Onda P:** una vez localizado un R, se define una ventana aproximada de 120–200 ms antes del pico. Dentro de esa ventana se busca el máximo local, asociado a la onda P en ritmos sinusales típicos.
- **Puntos Q y S:** alrededor de cada R se exploran ventanas cortas (por ejemplo ~ 60 ms antes y después) en busca de mínimos locales, que se identifican como Q (antes de R) y S (después de R). Esto permite estimar la duración del complejo QRS.
- **Onda T:** se define una ventana de 150–400 ms después de R (o S) y se busca el máximo local en ese intervalo, asociado a la onda T.

El resultado de esta etapa es un conjunto de marcas temporales por latido (tiempos de P, Q, R, S, T y límites aproximados), que servirán de base para el cálculo de los intervalos clínicos de interés.

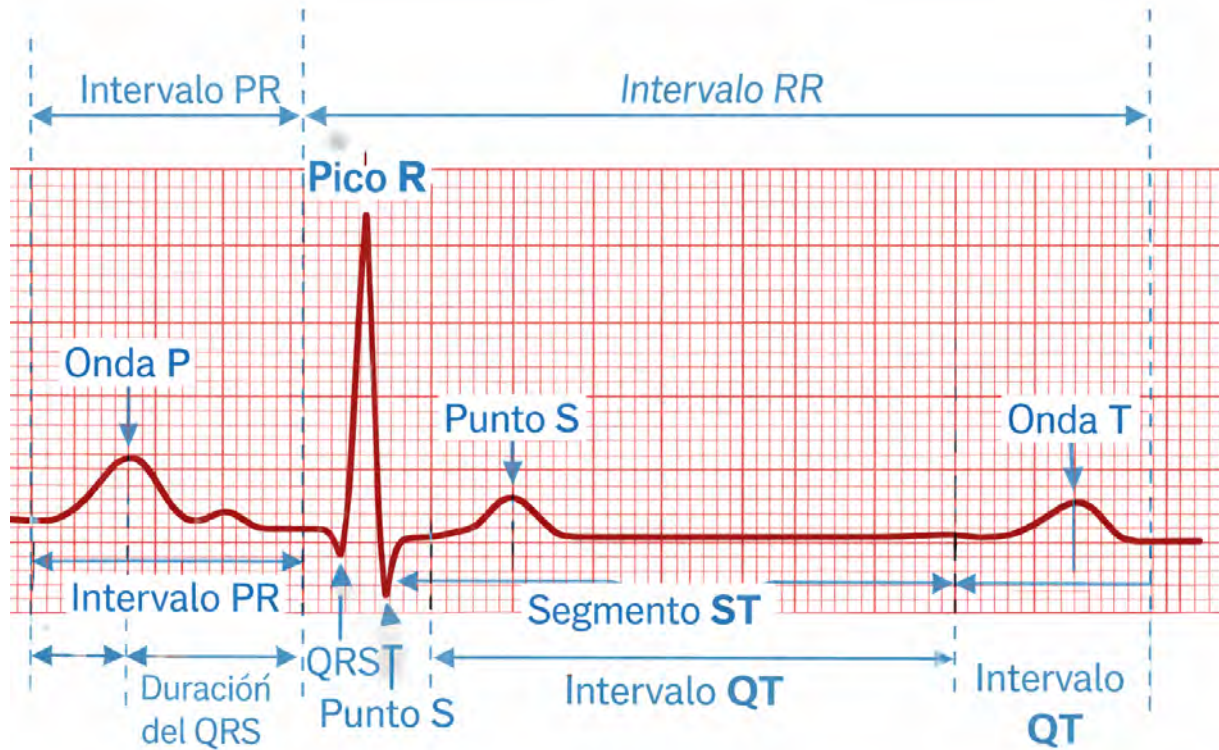


Figura 5.5: Ejemplo ilustrativo de un latido ECG con papel milimétrico, mostrando la localización de P, Q, R, S y T, así como la medición de intervalos PR, QRS, QT, ST y RR.

Cálculo de intervalos y derivados

Con las marcas temporales definidas (Sección 5.3), el módulo calcula los intervalos y parámetros derivados que se envían al servidor y, posteriormente, al modelo de aprendizaje automático. En general, si t_a y t_b son dos marcas temporales en segundos, el intervalo en milisegundos se obtiene como:

$$\Delta t_{\text{ms}} = 1000 (t_b - t_a).$$

La Figura 5.5 ilustra visualmente estas mediciones sobre papel milimétrico [47].

- **Intervalo RR:** se calcula como la diferencia temporal entre picos R consecutivos. A partir del valor promedio de estos intervalos se estima la frecuencia cardíaca:

$$\text{BPM} = \frac{60}{\text{RR}}.$$

Por ejemplo, un intervalo RR promedio de 0.8 s corresponde aproximadamente a 75 latidos por minuto.

- **PR, QRS, QT y ST:** se obtienen a partir de diferencias temporales entre marcas (ver Figura 5.5):
 - PR: desde el inicio de la onda P hasta el inicio de la onda Q,
 - QRS: desde Q hasta el final de S (punto J),
 - QT: desde Q hasta el final de T,
 - ST: desde el final de S (J) hasta el inicio de T (según la definición adoptada).

Estos intervalos se expresan en milisegundos y se comparan con rangos de referencia clínicos [47].

- **QT corregido (QTc):** El intervalo QT depende de la frecuencia cardíaca, por lo que se aplica una corrección para facilitar su interpretación clínica. Existen múltiples fórmulas descritas en la literatura (Bazett, Fridericia, Framingham y Hodges, entre otras) [48]. En este trabajo se emplea la corrección de Bazett:

$$QTc = \frac{QT}{\sqrt{RR}}$$

La elección de esta fórmula se debe a que es común en la práctica clínica diaria y en sistemas convencionales; además, su uso fue recomendado explícitamente por el experto clínico asesor del proyecto.

- **Ritmo y anomalías básicas:** Con base en reglas clínicas simples (por ejemplo: PR > 200 ms sugiere posible bloqueo AV de primer grado; QRS > 120 ms sugiere QRS ancho; BPM < 60 sugiere bradicardia; BPM > 100 sugiere taquicardia), se genera una primera etiqueta automática de ritmo y posibles anomalías.

Todos los parámetros se calculan a partir de intervalos promediados sobre registros de aproximadamente 30 s, lo que reduce la influencia de latidos aislados con morfología atípica, artefactos puntuales o ruido residual en la señal.

Persistencia local y envío al servidor

Al finalizar cada estudio, el módulo de captura conserva una copia local de la información y, a continuación, la sincroniza con el servidor web. La persistencia local y el envío se organizan de la siguiente manera:

- **CSV crudo y filtrado:** se generan dos archivos CSV:
 - uno con la señal cruda tal como llega de la placa,
 - otro con la señal filtrada (una columna, 200 Hz, 30 s).

Ambos se nombran incorporando el folio y la marca de tiempo.

- **PNG con malla clínica:** se crea una imagen PNG donde la señal filtrada se dibuja sobre una malla equivalente a 1 mV en el eje vertical y 200 ms en el eje horizontal. Esta imagen se aproxima a la apariencia del papel milimetrado del electrocardiograma y facilita la revisión por parte del médico.
- **JSON con listas RR y características:** además de los valores promedio (RR, PR, QRS, QT, ST, BPM, QTc, etc.), se genera un JSON que incluye la lista de intervalos RR detectados y otras series derivadas, útil para análisis posteriores o para recalculos si se actualiza el algoritmo.

El envío al servidor se realiza en tres pasos principales:

1. **Obtención y uso de un folio único.** El sistema solicita o construye un folio de 5 dígitos que identifica de forma única el estudio. Este folio se usa en los nombres de archivo, en la base de datos y en las URLs de consulta.
2. **Envío de características.** Las características calculadas (intervalos, BPM, QTc, ritmo, anomalías) se envían mediante una petición HTTP POST a un script del servidor, por ejemplo `actualizar_intervalos.php`, que valida los datos y los inserta en la tabla correspondiente de MySQL/MariaDB.
3. **Envío de archivos CSV/PNG.** Los archivos generados (CSV crudo, CSV filtrado, PNG) se envían con un POST de tipo `multipart/form-data` a otro script, por ejemplo `save_data.php`, que los guarda en el sistema de archivos del servidor y registra sus rutas en la base de datos.
4. **Apertura de la vista web.** Finalmente, el programa puede abrir automáticamente en el navegador la página de visualización, por ejemplo:

`visualizar_grafica.php?folio=...`

donde el médico podrá ver la traza, las características y, cuando esté disponible, el diagnóstico automático del modelo.

De esta forma, cada estudio pasa sin interrupciones desde la captura local hasta la plataforma web, quedando listo para ser evaluado por el cardiólogo y, en su caso, incorporado al conjunto de datos de entrenamiento del sistema de diagnóstico asistido.

5.4. Componente web (PHP + MySQL + Plotly)

Mientras que el módulo en Python se encarga de “escuchar” al corazón del paciente y extraer las características, el componente web es el encargado de organizar toda esa información, guardarla de forma segura y presentarla al médico de una manera clara. Esta parte del sistema se desarrolló con **PHP** (lógica del servidor), **MySQL/MariaDB** (base de datos) y **Plotly** (gráficas interactivas en el navegador).

La idea es que el personal clínico, al ingresar a la plataforma, pueda:

- buscar un estudio por folio o por fecha,
- ver la traza ECG con malla clínica,
- revisar las características calculadas por el sistema,
- consultar la propuesta de diagnóstico de la IA,
- y registrar, en el mismo lugar, su diagnóstico definitivo.

Autenticación y control de acceso

Dado que se trabaja con información sensible de pacientes, la plataforma web no es pública. El acceso se gestiona mediante un mecanismo de autenticación que valida las credenciales de usuario y asigna permisos en función de **roles** predefinidos. Para este fin, el sistema implementa un esquema de autenticación basado en *JSON Web Tokens* (JWT), lo que permite una gestión segura y controlada de las sesiones.

Una vez autenticado, cada usuario recibe un token firmado que se utiliza para autorizar las peticiones a la API y restringir el acceso a los recursos según su rol. Los roles definidos en el sistema son los siguientes:

- **Operador:** puede consultar estudios, visualizar la señal electrocardiográfica, revisar características básicas y marcar si un registro requiere repetición, pero no puede modificar diagnósticos.
- **Médico:** además de lo anterior, puede editar o registrar el diagnóstico clínico definitivo, marcar la condición del estudio (Normal/Anormal) y añadir observaciones clínicas.
- **Administrador:** puede gestionar usuarios, revisar registros de actividad y habilitar o deshabilitar funciones avanzadas del sistema (por ejemplo, iniciar un nuevo proceso de entrenamiento del modelo).

Cada rol cuenta con permisos diferenciados para ver, editar, exportar información o ejecutar procesos internos. Este enfoque asegura que únicamente el personal autorizado pueda tomar decisiones clínicas o acceder a información sensible, mientras que otros perfiles técnicos pueden participar en tareas de mantenimiento y soporte sin comprometer la confidencialidad de los datos de los pacientes.

El uso de JWT facilita además la integración con otros servicios institucionales, mejora la escalabilidad del sistema y permite un control más fino de las sesiones, aspectos relevantes en un entorno clínico donde la seguridad y la trazabilidad del acceso son fundamentales.

API PHP y contratos de datos

La comunicación entre el módulo de captura en Python y la plataforma web se realiza a través de una pequeña **API en PHP**, organizada en varios scripts que actúan como “puertas de entrada” al sistema. Cada uno de estos scripts define claramente qué datos espera recibir y qué respuesta devolverá, lo que constituye un *contrato de datos* entre ambas partes.

Guardar archivos (save_data.php). Este endpoint se encarga de recibir los archivos generados al finalizar cada captura.

- **Entrada:** Se envían, mediante un POST de tipo `multipart/form-data`, los siguientes elementos:
 - folio del estudio,
 - archivos CSV (señal cruda y señal filtrada),
 - archivo PNG con la traza en malla clínica,
 - metadatos básicos (sexo, edad, fecha/hora de captura).
- **Validación:** El script verifica que los archivos tengan el tipo y tamaño esperados, calcula un hash **SHA-256** para cada archivo (opcional, para integridad), crea una carpeta organizada por fecha y genera nombres de archivo que combinan el folio con un sello de tiempo. De este modo se evita sobrescribir archivos y se facilita el orden.
- **Salida:** La respuesta es un objeto JSON que indica si la operación fue exitosa e incluye las rutas relativas de los archivos dentro del servidor.

Actualizar características (actualizar_intervalos.php). Este endpoint recibe las características ya calculadas en Python y las registra en la base de datos.

- **Entrada:** Los datos se envían típicamente como `application/x-www-form-urlencoded` o JSON e incluyen:
 - folio,
 - `rr_interval`, `pr_interval`, `qrs_duration`,
 - `qt_interval`, `st_interval`,
 - `frecuencia_cardiaca`, `p_duration`, `qtc`,
 - conteos de ondas (`onda_p/q/r_count`),
 - etiquetas como `ritmo`, `anomalias`,
 - y un indicador `requiere_revision` para marcar registros dudosos.
- **Salida:** El script actualiza la fila correspondiente en la tabla `registros_ecg` y devuelve un JSON confirmando el UPDATE, junto con un mensaje de éxito o de error.

Entrenamiento (obtener_datos_entrenamiento.php / registrar_entrenamiento.php). Estos endpoints permiten que un script en Python descargue datos etiquetados para entrenar el modelo y, posteriormente, suba los resultados del entrenamiento.

- **GET (obtener_datos_entrenamiento.php):** Devuelve las filas de la base de datos que ya cuentan con una etiqueta clínica (Normal/Anormal) y que están autorizadas para entrenamiento. Cada fila incluye las características calculadas (RR, PR, QRS, QT, ST, BPM, QTc, `p_duration`, etc.) y la condición.

- **POST (`registrar_entrenamiento.php`):** Una vez entrenado el modelo, Python envía:
 - un `metrics_json` con las características de validación (ACC, Recall, AUC, etc.),
 - y archivos adjuntos, como el paquete `.pkl` del modelo y figuras de ROC o del árbol de decisión.

El servidor guarda estos artefactos, de modo que quede constancia de cómo se entrenó el modelo en cada versión.

Predicción (`obtener_datos_para_prediccion.php` / `registrar_predicciones.php`). Finalmente, estos endpoints gestionan el flujo de predicción automática.

- **GET (`obtener_datos_para_prediccion.php`):** Entrega al script en Python los registros *pendientes* de clasificación, es decir, aquellos que ya tienen características pero aún no cuentan con diagnóstico por IA. Cada elemento incluye el `folio` y las columnas de características esperadas por el modelo.
- **POST (`registrar_predicciones.php`):** Tras aplicar el modelo, Python envía una lista de objetos con la forma `{folio, clase, prob_anormal}`. El servidor registra para cada folio la clase propuesta (Normal/Anormal), la probabilidad de anomalía y, si se desea, la versión del modelo que realizó la predicción.

Base de datos y esquema

El corazón de la parte web es la tabla `registros_ecg`, donde cada fila representa un estudio asociado a un `folio`. Esta tabla concentra tanto la información demográfica básica como las características, los diagnósticos y las rutas a los archivos asociados.

A grandes rasgos, los campos se agrupan de la siguiente manera:

- **Demográficos:** sexo, edad, fecha y hora del estudio. Estos datos permiten describir la población analizada y, en un futuro, estudiar patrones por grupo de edad o por sexo.
- **Características de la señal:** campos como `frecuencia_cardiaca`, `rr_interval`, `pr_interval`, `qrs_duration`, `qt_interval`, `st_interval`, `p_duration`, `qtc`, `ritmo`, `anomalias`, así como `rr_intervals` (JSON) y conteos de ondas P/Q/R. Estos valores corresponden al resumen numérico del ECG.
- **Diagnóstico:** se distinguen varios campos:
 - `diagnostico`: texto generado por la IA (por ejemplo, “Patrón compatible con taquicardia sinusal”),
 - `diagnostico_doctor`: diagnóstico clínico registrado por el cardiólogo,
 - `condicion`: etiqueta simplificada Normal/Anormal que resume el estado del estudio,

- **requiere_revision:** bandera que indica si el registro debe revisarse con especial cuidado (por ejemplo, por calidad dudosa o discrepancias entre IA y médico).
- **Archivos:** rutas a los CSV (crudo y filtrado) y a las imágenes PNG de la traza. Esto permite abrir, desde la propia interfaz web, tanto la visualización interactiva como los archivos originales.

Tabla 5.1: Esquema resumido de `registros_ecg` (campos clave)

Campo	Descripción
folio (PK lógico)	Identificador único por captura
sexo, edad, fecha	Metadatos básicos del estudio
frecuencia_cardiaca	BPM promedio en la ventana de registro
rr_interval, pr_interval,	Intervalos medios (s)
qrs_duration, qt_interval,	
st_interval	
p_duration, qtc	Derivados clínicos (s)
onda_p/q/r_count	Conteo de ondas detectadas por el algoritmo
ritmo, anomalias	Interpretación automática básica de la señal
diagnostico, diagnostico_doctor,	Etiquetas de IA y diagnóstico clínico
condicion	
rr_intervals (JSON)	Serie completa de intervalos RR por latido

Este diseño permite que el sistema crezca con el tiempo: se pueden añadir nuevos campos (por ejemplo, presión arterial o saturación de oxígeno) manteniendo la compatibilidad con los scripts ya existentes, siempre que se actualicen también los contratos de datos de la API.

Visualización web (Plotly)

La interfaz web es el rostro visible del sistema para el médico. Está diseñada para que, al abrir un estudio, la persona vea algo familiar: una traza ECG sobre una malla similar a la del papel milimetrado y con marcadores visuales que facilitan la ubicación de ondas y el análisis de intervalos clínicos. En particular, la convención de malla e intervalos reportados es consistente con el ejemplo ilustrativo de la Fig. 5.5.

Con **Plotly** se genera una gráfica interactiva en la que:

- la **señal filtrada** se dibuja con una escala clínica equivalente a 25 mm/s (200 ms por cuadro grande) en el eje horizontal y 10 mm/mV (1 mV por 10 mm) en el eje vertical, como en papel ECG; esta escala es la que permite medir visualmente duraciones y amplitudes (véase Fig. 5.5);
- se superponen **marcadores** que indican la posición de las ondas P, los complejos QRS y las ondas T (y, cuando aplica, puntos Q y S), facilitando la revisión visual y la validación clínica de las detecciones automáticas;

- el usuario puede hacer *zoom*, desplazar la vista, y activar o desactivar capas (por ejemplo, mostrar u ocultar marcadores e intervalos).

Junto a la gráfica se presenta un panel con las **características** del registro (\overline{RR} , PR, QRS, QT, ST, QTc, BPM, ritmo y anomalías) y el diagnóstico actual:

- la condición estimada por la IA (Normal/Anormal), con la probabilidad asociada,
- el diagnóstico clínico registrado por el médico,
- y un campo para marcar si el estudio `requiere_revision`.

La interfaz incluye controles para:

- **exportar** la traza y las características a formatos PDF o CSV, lo que permite incorporarlas a expedientes electrónicos o informes;
- **marcar** o desmarcar el indicador `requiere_revision` cuando el médico lo considere conveniente;
- navegar entre estudios de un mismo paciente o filtrar por fecha o por condición (por ejemplo, mostrar solo los registros anormales).

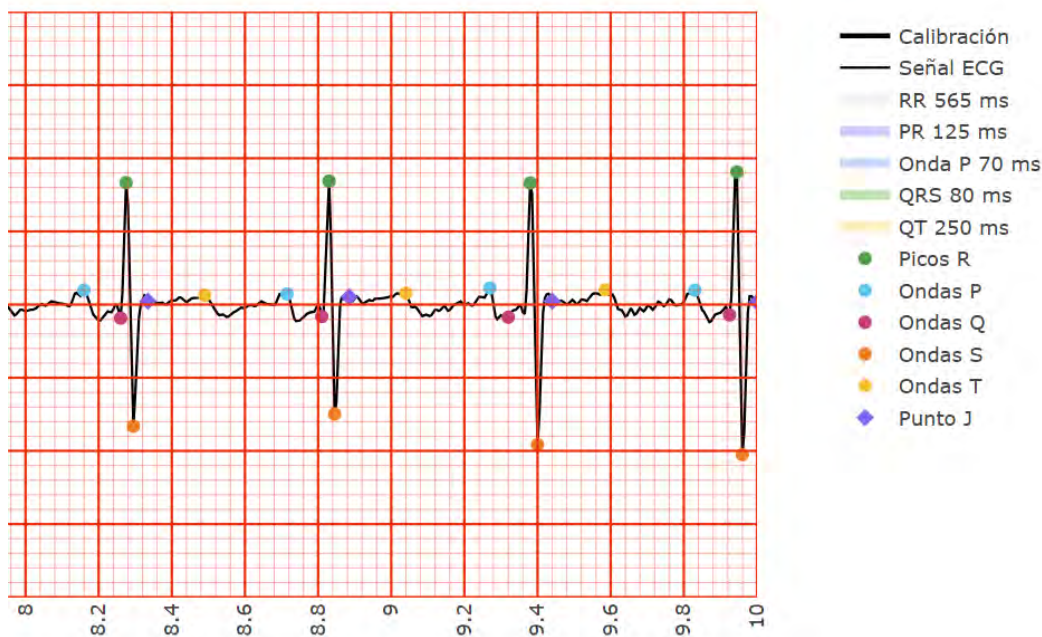


Figura 5.6: Visualización web de un registro ECG con malla clínica y marcadores para ondas P/QRS/T. El panel lateral muestra las características calculadas y los diagnósticos registrados.

De esta forma, el componente web se convierte en un espacio de trabajo compartido entre la ingeniería y la medicina: la parte técnica se encarga de procesar y organizar los datos, mientras que la interfaz permite que el especialista valide visualmente las mediciones, interprete el estudio y deje constancia de su diagnóstico en el sistema.

Módulo de notificación y carga de ECG de 12 derivaciones

Tras implementar la plataforma web descrita en la Sección 5.4, se añadió un módulo que permite solicitar y gestionar electrocardiogramas de 12 derivaciones cuando el algoritmo de clasificación detecta una condición *Anormal*. El propósito es obtener información adicional (ECG estándar de 12 derivaciones) para que el Dr. Juan Manuel López Quijano pueda revisar el caso y, con ello, apoyar la validación clínica del resultado arrojado por el sistema. Cabe aclarar que este ECG estándar de 12 derivaciones **debe obtenerse de manera externa** mediante un electrocardiógrafo clínico convencional (u otro servicio/estudio hospitalario equivalente), ya que **el sistema propuesto no genera registros de 12 derivaciones**; únicamente solicita, recibe y gestiona el documento para su revisión.

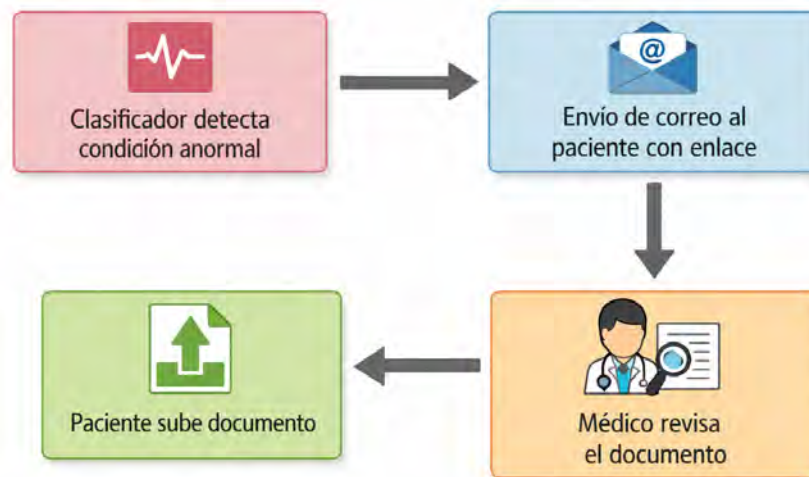


Figura 5.7: Flujo del módulo de notificación y carga de documentos: (1) el clasificador detecta una condición anormal; (2) el sistema envía un correo con un enlace a la página de carga; (3) el paciente sube un ECG de 12 derivaciones; y (4) el médico revisa el documento y registra su revisión clínica.

Notificación por correo electrónico El script `enviar_solicitud_archivo.php` selecciona los registros cuya condición es **Anormal**, comprueba que el paciente tenga un correo electrónico registrado y que el campo `requiere_revision` esté en cero. Para cada caso envía

un mensaje mediante PHPMailer con un enlace directo al formulario de carga. El correo se redacta en un tono neutral, sin alarmar, y se enfoca en solicitar el ECG de 12 derivaciones para revisión. Al finalizar, el script actualiza `requiere_revision` para evitar notificaciones duplicadas.

Carga de archivos La página `subir_documento.php` corresponde al formulario donde el usuario puede cargar su archivo en PDF o imagen. Esta vista recibe como parámetro el *folio*, valida el tipo MIME permitido (PDF, JPEG o PNG), genera un nombre único (folio + sello temporal) y almacena el archivo en la carpeta `uploads`. De forma paralela registra la carga en la tabla `documentos_subidos`, asociando folio, nombre del archivo, tipo y fecha de subida. En pantalla se aclara al paciente que esta solicitud no significa necesariamente un problema grave, sino una revisión adicional por parte del especialista.

Visualización y revisión de documentos Para usuarios con rol de *médico* o *administrador* se ofrece la vista `documentos_12d.php`, donde se listan los documentos subidos (por fecha, folio y paciente). La tabla muestra un estado de “pendiente” con base en `requiere_revision`. Desde aquí, el Dr. Juan Manuel López Quijano, **u otro médico que posteriormente pueda agregarse**, puede abrir el PDF/imagen para revisarlo y posteriormente marcar el registro como atendido, cerrando el ciclo de revisión clínica.

Encuesta de retroalimentación para médicos Además del módulo de revisión, se añadió una encuesta breve (`encuesta_doctor.php`) para que el personal médico evalúe la experiencia de uso del sistema. El formulario solicita opcionalmente el nombre del participante e incluye preguntas de escala (1–5) sobre satisfacción general y facilidad de uso, además de campos abiertos para comentarios y sugerencias. Las respuestas se almacenan mediante `guardar_encuesta_doctor.php` en la tabla `encuesta_doctor`. Es importante recalcar que esta encuesta no evalúa al paciente, sino la utilidad, claridad y funcionamiento general de la plataforma como herramienta de apoyo, permitiendo orientar mejoras futuras en la interfaz y el flujo de trabajo.

Mensaje de notificación al paciente El sistema envía al paciente un correo electrónico automático cuando se detecta un registro que requiere revisión adicional. El mensaje se redacta en un lenguaje claro y no alarmante, informando que el resultado no implica necesariamente un problema de salud, sino la necesidad de una segunda revisión clínica.

El correo incluye el folio del registro y un enlace directo a la plataforma para cargar un electrocardiograma de 12 derivaciones en formato PDF o imagen. De esta manera, se facilita la participación del paciente y se completa el flujo de seguimiento médico sin necesidad de contacto directo adicional.

5.5. Sesiones de revisión clínica e integración hospitalaria

Una parte fundamental de este proyecto fue sacar el sistema del entorno de laboratorio y contrastarlo con la realidad clínica. Para ello se llevaron a cabo varias sesiones de validación en el **Hospital Central de San Luis Potosí**, en coordinación con el jefe de Cardiología y el equipo técnico del servicio. Estas sesiones permitieron ajustar parámetros, comparar resultados contra equipos de referencia y construir, en conjunto, las etiquetas clínicas que el modelo utilizaría para aprender.

De manera resumida, los objetivos principales de estas reuniones fueron:

1. **Ajustar parámetros de captura y filtrado.** En una primera etapa se conectó la placa *OpenBCI Ganglion* a pacientes reales, en paralelo con el electrocardiógrafo clínico. Se comparó en tiempo real la traza obtenida por el prototipo con la del equipo hospitalario estándar. A partir de esta comparación se ajustaron parámetros como la frecuencia de muestreo efectiva, el rango de filtrado (0.5–45 Hz) y la configuración del filtro *notch* a 60 Hz, hasta obtener una morfología de P–QRS–T lo más cercana posible a la de referencia.
2. **Verificar latencias y amplitudes frente a equipo clínico de referencia.** En siguientes sesiones se revisaron con el cardiólogo las medidas de intervalos (PR, QRS, QT, QTc) producidas por el sistema y se contrastaron con las mediciones realizadas directamente en el equipo clínico o sobre impresiones de los trazos. El objetivo fue confirmar que las diferencias en latencia y amplitud se mantenían dentro de márgenes aceptables, teniendo en cuenta la diferencia de hardware, y detectar posibles sesgos sistemáticos (por ejemplo, una tendencia a subestimar ligeramente el QT o a detectar menos latidos en presencia de ruido).
3. **Consolidar `diagnostico_doctor` y comentarios clínicos para el etiquetado de entrenamiento.** Cada registro revisado se acompañó de un diagnóstico clínico escrito por el cardiólogo responsable, así como de observaciones relevantes (por ejemplo: “taquicardia sinusal”, “bloqueo de rama derecha”, “registro de calidad limitada por artefactos de movimiento”). Estas anotaciones se incorporaron al campo `diagnostico_doctor` en la base de datos y sirvieron como *ground truth* para etiquetar los casos como *Normales* o *Anormales*, lo que resultó esencial para entrenar el modelo de aprendizaje supervisado.

Las Figuras 5.8 y 5.9 ilustran dos de estas sesiones de trabajo conjunto, en las que se revisaron casos patrón, se realizaron pruebas en tiempo real con electrodos y se discutieron criterios de interpretación.



Figura 5.8: Primera sesión de revisión con el equipo de Cardiología del Hospital Central. Se analizaron casos patrón y se realizaron pruebas en tiempo real para comparar la traza del prototipo con el equipo clínico de referencia.



Figura 5.9: Segunda sesión de ajuste fino de parámetros (frecuencia de muestreo efectiva, filtros, umbrales de detección) y revisión de la concordancia clínica entre el sistema propuesto y el electrocardiógrafo hospitalario.

Estas interacciones no solo fortalecieron la validez técnica del sistema, sino que también permitieron que el equipo clínico se apropiara del proyecto, aportando su experiencia para guiar las decisiones de diseño y señalando los aspectos más importantes desde el punto de vista del paciente.

5.6. Preparación del conjunto de datos

Antes de entrenar cualquier modelo de aprendizaje automático es necesario contar con un conjunto de datos coherente, limpio y bien etiquetado. En clasificación de ECG, esto significa

que para cada estudio se debe disponer, por un lado, de información numérica que describa la señal y, por otro, de un diagnóstico clínico de referencia emitido por un especialista.

De forma general, cada registro del conjunto de datos representa a un paciente (o a un estudio de ECG) y está compuesto por:

- **Información de la señal electrocardiográfica**, ya sea en forma de señal cruda, señal filtrada o latidos previamente segmentados.
- **Características numéricas** derivadas del ECG, como duraciones de intervalos, segmentos y frecuencia cardíaca.
- **Etiqueta clínica** que indica si el estudio se considera *Normal* o *Anormal* de acuerdo con la valoración médica.

Este esquema es típico en problemas de apoyo al diagnóstico: la señal continua se resume en un conjunto de números con significado fisiológico y esos números se asocian a una decisión clínica que el modelo tratará de aprender.

Fuentes y estructura de la información

En el contexto de los ECG, el conjunto de datos suele construirse a partir de tres tipos de fuentes:

- **Señal de ECG cruda y/o filtrada.**
A partir de la señal registrada se pueden repetir o mejorar los algoritmos de procesamiento en el futuro. Aunque el modelo de clasificación no usa directamente todas las muestras de la señal, conservar esta información permite recalcular características si se ajustan los filtros o los detectores de ondas.
- **Métricas por registro.**
Para cada estudio se calculan valores promedio de los principales intervalos y segmentos: \overline{RR} , \overline{PR} , \overline{QRS} , \overline{QT} , \overline{ST} , así como la frecuencia cardíaca (BPM) y el QT corregido (QTc), entre otros. Estas cantidades condensan la información más relevante de la señal en unas pocas variables fáciles de interpretar.
- **Etiqueta clínica de referencia.**
Cada registro se marca como *Normal* o *Anormal* siguiendo el criterio de un médico especialista. Esta etiqueta es la “verdad terreno” (*ground truth*) que el modelo intenta reproducir. Sin etiquetas confiables no es posible entrenar ni evaluar un sistema de apoyo al diagnóstico.

A partir de estos elementos se construye una tabla en la que cada fila corresponde a un estudio de ECG y cada columna representa una característica o un metadato relevante.

Limpieza y codificación de etiquetas

Una vez reunida la información, es necesario llevar a cabo un proceso de *limpieza de datos*. De forma general, este proceso incluye:

- **Homogeneizar nombres y unidades.**
Se unifican los nombres de las variables (por ejemplo, usar siempre “QTc” y no mezclar “QTc” con “qtc”) y se revisa que los intervalos estén expresados en las mismas unidades (milisegundos, segundos, etc.).
- **Tratar valores faltantes o no válidos.**
Es frecuente encontrar registros con medidas incompletas o valores numéricamente inestables. En estos casos se eliminan los estudios con información insuficiente o se aplican estrategias de imputación, según la calidad y cantidad de datos disponibles.
- **Codificar la etiqueta de manera consistente.**
Para que el modelo pueda trabajar con las etiquetas, es habitual traducirlas a valores binarios:

$$y = \begin{cases} 1, & \text{si el estudio se considera Anormal,} \\ 0, & \text{si el estudio se considera Normal.} \end{cases}$$

Esta codificación binaria facilita el uso de la mayoría de los algoritmos de clasificación supervisada.

El resultado de esta fase es un conjunto de datos donde las columnas están claramente definidas, las unidades son comparables y las etiquetas siguen un esquema uniforme.

Características empleadas en clasificación de ECG

El vector de características utilizado como entrada del modelo se construye a partir de medidas clínicamente significativas de la señal ECG. Un conjunto típico de variables incluye:

- **Intervalo RR:** tiempo entre latido y latido, relacionado con la frecuencia cardíaca y la variabilidad del ritmo.
- **Intervalo PR:** refleja la conducción auriculoventricular y es sensible a ciertos tipos de bloqueos.
- **Duración del QRS:** representa la despolarización ventricular; su ensanchamiento puede indicar trastornos de conducción.
- **Intervalo QT:** abarca la actividad eléctrica ventricular completa (despolarización y repolarización).
- **Segmento ST:** corresponde a la fase inicial de repolarización ventricular y es clave en la detección de isquemia.
- **Frecuencia cardíaca (BPM):** resume el ritmo global del paciente.

- **QT corregido (QTc):** ajuste del intervalo QT en función de la frecuencia cardíaca, ampliamente utilizado en la práctica clínica.
- **Duración aproximada de la onda P:** asociada a la actividad eléctrica auricular.

Estas variables pueden denotarse, por ejemplo, como `rr_interval`, `pr_interval`, `qrs_duration`, `qt_interval`, `st_interval`, `frecuencia_cardiaca`, `qtc` y `p_duration`, formando un vector de entrada compacto e interpretable.

Aunque en la práctica algunos estudios pueden carecer de una o varias de estas medidas (por ejemplo, cuando no es posible estimar con fiabilidad cierta onda), la idea general es que el modelo se alimente de un conjunto de características que, además de ser informativas desde el punto de vista estadístico, tengan un significado fisiológico claro para el cardiólogo.

En los siguientes apartados se describen los modelos de aprendizaje automático que se utilizarán para relacionar estas características con la etiqueta *Normal/Anormal*, prestando especial atención a los árboles de decisión y a su aplicación en el contexto de las señales ECG.

5.7. Módulo de entrenamiento y predicción (Python)

Además del módulo de captura en tiempo real, el sistema cuenta con un componente desarrollado en **Python** que se encarga de **entrenar** periódicamente el modelo de clasificación y **predecir** la condición (*Normal / Anormal*) de los registros que aún no han sido evaluados por la IA. Este módulo está integrado dentro del script principal `ecg_ai_unificado.py` y funciona como el “cerebro estadístico” del sistema, comunicándose con la base de datos a través de la API PHP descrita en la Sección 5.4. La Figura 5.10 resume el flujo general del proceso, incluyendo la generación del *bundle* (paquete reproducible con el modelo entrenado, hiperparámetros y umbral de decisión).

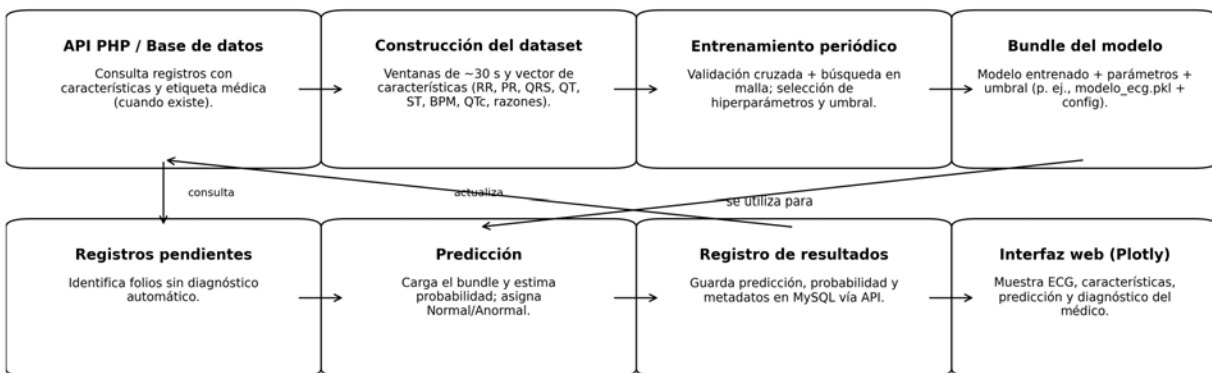


Figura 5.10: Diagrama de bloques del módulo `ecg_ai_unificado.py`: construcción del conjunto de entrenamiento a partir de la API, entrenamiento periódico con validación cruzada y búsqueda en malla, empaquetado del modelo (*bundle*) y predicción de registros pendientes con registro de resultados para su visualización en la interfaz web.

Cada registro de entrenamiento corresponde a una ventana de aproximadamente **30 segundos** de señal ECG filtrada, sobre la cual ya se han calculado las características descritas

en el Capítulo 3 (intervalos RR, PR, QRS, QT, ST, frecuencia cardíaca, QTc, recuentos de ondas y razones QT/RR y ST/QT, entre otras).

Entrenamiento supervisado

El procedimiento de entrenamiento se ha diseñado para ser lo más automático y reproducible posible, siguiendo la metodología del Capítulo 4. De forma resumida, el flujo que implementa la función `train_model()` es el siguiente:

1. Obtención de datos etiquetados.

El script realiza una petición GET al endpoint `obtener_datos_entrenamiento.php`. Este servicio devuelve un conjunto de filas de la tabla `registros_ecg` que ya cuentan con etiqueta clínica (campo `condicion`, Normal/Anormal) asignada por el cardiólogo, así como con todas las características necesarias: intervalos RR, PR, QRS, QT, ST, frecuencia cardíaca, QTc, número de ondas P/Q/R detectadas, oxigenación, peso, perímetro de cintura, edad y las razones derivadas `qt_rr_ratio` y `st_qt_ratio`. Los datos se cargan en un `DataFrame` de `pandas`.

2. Preprocesamiento y selección de características.

Todas las columnas numéricas se convierten explícitamente a tipo `float`, se depuran registros con valores imposibles (por ejemplo, divisiones entre cero) y se imputan los campos opcionales con su mediana o moda según corresponda. A partir de ahí se construye el vector final de entrada:

$$X = \{\text{rr_interval}, \text{pr_interval}, \text{qrs_duration}, \text{qt_interval}, \\ \text{st_interval}, \text{frecuencia_cardiaca}, \text{qtc}, \text{onda_p_count}, \\ \text{onda_q_count}, \text{onda_r_count}, \text{oxigenacion}, \text{peso}, \\ \text{cintura}, \text{edad}, \text{qt_rr_ratio}, \text{st_qt_ratio}\}.$$

Para reducir el impacto de valores extremos se aplica un recorte (*winsorization*) entre los percentiles 1% y 99% de cada característica.

3. Balanceo del conjunto de entrenamiento.

Puesto que el número de estudios normales y anormales puede variar a lo largo del tiempo, el módulo permite diferentes políticas de balanceo a través del parámetro `PER_CLASS`. En la configuración utilizada para los resultados presentados en esta tesis se eligió un esquema estrictamente balanceado, con **45 registros normales y 45 anormales** (`PER_CLASS = 45`). De esta manera se evita que el modelo se incline de forma artificial hacia la clase mayoritaria y se facilita la interpretación clínica de las características.

4. Entrenamiento del clasificador basado en árboles.

Sobre el conjunto balanceado se entrena un modelo de tipo bosque aleatorio (*Random Forest*) construido a partir de árboles de decisión tipo CART. El script utiliza `GridSearchCV` con validación cruzada estratificada (8 pliegues) para explorar diferentes

combinaciones de hiperparámetros, como el número de árboles (`n_estimators`), la profundidad máxima (`max_depth`) y el tamaño mínimo de las hojas (`min_samples_leaf`). La métrica objetivo para seleccionar la mejor configuración es el **F₁ macro**, lo que obliga al modelo a desempeñarse bien tanto en la clase Normal como en la clase Anormal.

Para el conjunto de datos disponible, la mejor configuración resultó en un bosque con profundidad máxima `max_depth = 6` y un total de `n_estimators = 600` árboles (entre otros parámetros menores), utilizando 8 pliegues de validación cruzada. Esta información se almacena en el campo `params` del bundle de modelo.

5. Selección del umbral de decisión.

A partir de las probabilidades *out-of-fold* (*OOF*) generadas durante la validación cruzada se exploran distintos umbrales de decisión. El módulo implementa varias políticas (`zouden`, `.acc`, `match_rec_acc`, `f1`), y en esta tesis se empleó la política `f1`, que selecciona el umbral que maximiza el F₁ macro. El umbral óptimo encontrado fue aproximadamente $t^* = 0,43$; es decir, se considera un estudio Anormal si la probabilidad estimada de anormalidad es mayor o igual a 0.43.

6. Construcción del bundle de despliegue.

Una vez elegido el mejor modelo y el umbral, el sistema entrena un clasificador final con todos los datos balanceados y construye un paquete (*bundle*) en formato `pickle`: `modelo_ecg.pkl`. Este bundle incluye:

- el estimador entrenado (`model`),
- el umbral t^* seleccionado (`threshold`),
- la lista ordenada de características de entrada (`feature_cols`),
- los hiperparámetros ganadores y la configuración de validación cruzada,
- el tamaño efectivo por clase (`per_class = 45`).

De esta forma es posible reutilizar exactamente el mismo modelo en la fase de predicción, sin necesidad de reentrenarlo cada vez.

7. Registro de características y figuras en el servidor.

Finalmente, la función `train_model()` genera varias figuras (curva ROC, curva Precisión-Recall, matriz de confusión, importancia de características y un árbol representativo del bosque) y empaqueta junto con el bundle un objeto `metrics_json` que resume las características clave. Todo ello se envía mediante una petición `POST` al endpoint `registrar_entrenamiento.php`, donde queda almacenado el historial de entrenamientos accesible desde la interfaz web.

Interpretabilidad: importancia de características y árbol representativo

Aunque internamente el clasificador es un bosque de múltiples árboles, el sistema genera dos salidas que ayudan a explicar las decisiones del modelo:

1. Importancia de características.

A partir de los coeficientes de importancia del *Random Forest* se calcula el peso relativo de cada variable en las decisiones del modelo. En este trabajo, dichas importancias corresponden a la contribución promedio de cada característica a la reducción de impureza (criterio de Gini) a lo largo de los árboles y se normalizan para facilitar su comparación. La Figura 6.3 muestra que las características con mayor aportación al modelo son: `pr_interval`, `qrs_duration`, `edad`, `qt_interval` y las razones derivadas `qt_rr_ratio` y `st_qt_ratio`.

Este resultado es consistente con la práctica clínica: el intervalo PR se asocia con la conducción auriculoventricular; la duración del QRS con la despolarización ventricular y alteraciones de conducción intraventricular; y las medidas relacionadas con QT (y su relación con RR) con fenómenos de repolarización. La edad, por su parte, puede actuar como un factor clínico acompañante que modula el riesgo de presentar hallazgos anormales en el conjunto analizado.

¿Por qué otras características aparecen con menor importancia? En primer lugar, varias variables pueden estar *correlacionadas* o ser *redundantes* (por ejemplo, una razón derivada puede capturar parte de la información de dos intervalos base); en estos casos, el modelo suele concentrar importancia en una de ellas y reducirla en otras sin que eso implique que sean irrelevantes clínicamente. En segundo lugar, algunas características pueden presentar menor variabilidad o menor capacidad de separación dentro de este conjunto específico (por tamaño muestral, distribución de clases o calidad de medición), por lo que aportan menos ganancia al separar Normal/Anormal. Finalmente, es importante señalar que la importancia por Gini puede sesgarse hacia variables con más puntos de corte posibles y no debe interpretarse como causalidad; por ello, estas importancias se usan como guía de auditoría clínica y se complementan con revisión del cardiólogo y, cuando se requiera, con métodos más robustos (p.ej., *permutation importance*).

2. Árbol de decisión representativo.

Para ofrecer una explicación aún más intuitiva, el sistema extrae uno de los árboles del bosque (recortado a profundidad 3) y lo dibuja como árbol de decisión representativo (Figura 6.9). Cada hoja del árbol corresponde a una regla del tipo “si se cumplen ciertas condiciones, entonces el estudio tiende a clasificarse como Normal o Anormal”. Al revisar las hojas del árbol, se observan patrones que, en su mayoría, resultan clínicamente plausibles:

- **Coherente:** combinaciones de `pr_interval` en rango y `qrs_duration` bajo tienden a inclinar la decisión hacia *Normal*, lo cual es consistente con conducción auriculoventricular y ventricular sin alteraciones marcadas.
- **Coherente:** aumentos del `pr_interval` (conducción AV más lenta) y/o `qrs_duration` por encima de un umbral tienden a incrementar la probabilidad de *Anormal*, lo cual coincide con la interpretación clínica de trastornos de conducción o complejos ventriculares ensanchados.

- **Coherente con cautela:** reglas que incorporan razones como `qt_rr_ratio` o `st_qt_ratio` suelen apuntar a cambios en repolarización; sin embargo, su lectura depende del contexto clínico y de la calidad del cálculo de intervalos (p. ej., presencia de ruido, variabilidad del RR o errores de detección de ondas).
- **Reglas a vigilar:** cuando el árbol utiliza variables clínicas complementarias (p. ej., edad u otras mediciones disponibles en el expediente como oxigenación), estas pueden reflejar correlaciones del conjunto de entrenamiento y no necesariamente una relación directa con el trazado ECG. En especial, si una hoja se apoya en pocas muestras (*samples* reducido en el nodo), la regla debe interpretarse como indicio y no como criterio clínico definitivo.

En síntesis, el árbol representativo funciona como una explicación aproximada del comportamiento del bosque: permite comunicar al cardiólogo *qué variables suelen disparar la decisión y en qué dirección*, pero las reglas deben validarse clínicamente, especialmente en aquellas hojas con baja proporción de muestras o con condiciones que no sean directamente atribuibles al ECG.

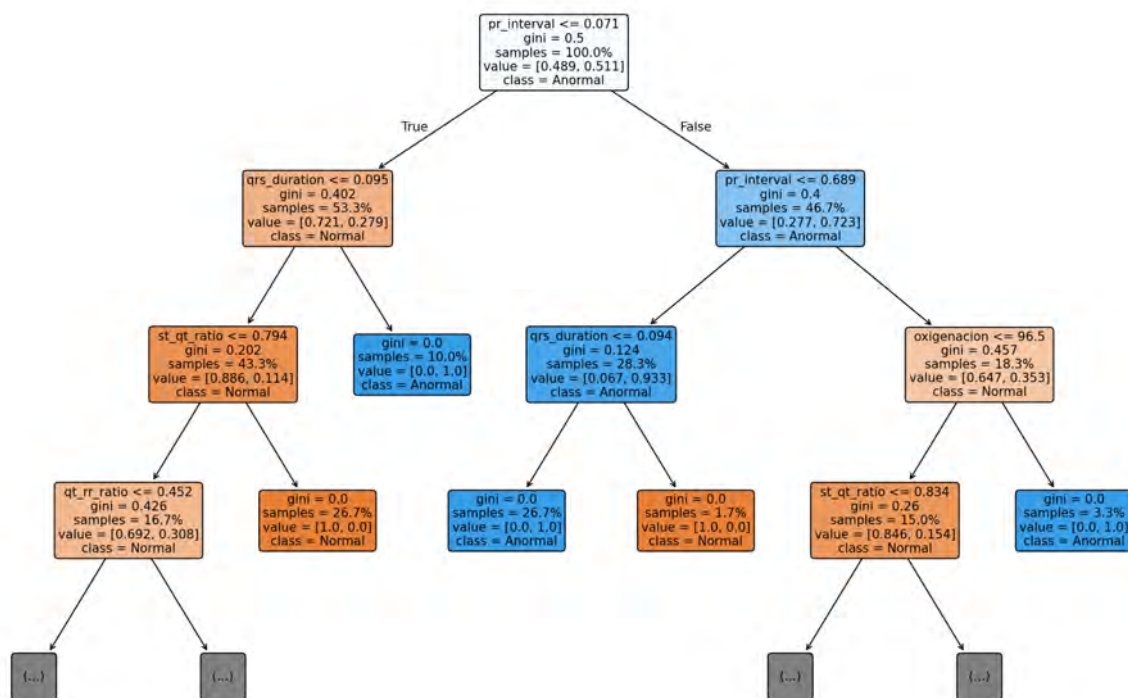


Figura 5.11: Árbol de decisión representativo extraído del bosque aleatorio utilizado en producción (profundidad limitada a tres niveles). Cada nodo muestra la característica de corte, el índice de Gini, la proporción de muestras y la clase predominante (Normal o Anormal).

Predicción automática en producción

El mismo módulo de Python se encarga de aplicar el modelo entrenado a los estudios que aún no han sido clasificados por la IA. Esta tarea se implementa en la función `predict_data()`, que puede invocarse desde el menú del programa o programarse para ejecución periódica. El ciclo de trabajo es el siguiente:

1. **Obtención de registros pendientes.**

El script realiza una petición GET al endpoint `obtener_datos_para_prediccion.php`. El servidor responde con una lista de estudios sin diagnóstico automático, incluyendo su `folio` y las características necesarias. Cada fila contiene los intervalos RR, PR, QRS, QT, ST, la frecuencia cardíaca y, cuando están disponibles, los valores de QTc, conteo de ondas, oxigenación, peso, cintura y edad.

2. **Reconstrucción del vector de entrada.**

Para cada estudio, Python reconstruye un diccionario de valores respetando el orden exacto de `feature_cols`. Cuando el modelo espera características derivadas (QTc, `qt_rr_ratio`, `st_qt_ratio`), estas se recalculan a partir de los intervalos base, de manera que la lógica de predicción sea coherente con la usada en el entrenamiento.

3. **Inferencia con el bundle.**

El bundle `modelo_ecg.pkl` se carga desde disco y se aplica sobre cada nuevo registro. El modelo devuelve una probabilidad de anormalidad $\hat{p} = \mathbb{P}(y = 1 | X)$ y, utilizando el umbral t^* almacenado, se determina la **clase propuesta** (*Normal* o *Anormal*). La probabilidad se conserva en el campo `prob_anormal`, lo que permite al médico valorar la confianza de la IA (por ejemplo, un estudio con 61 % de probabilidad de anormalidad se muestra en la interfaz como “Anormal (61 %)”).

4. **Registro de predicciones en la base de datos.** Con las predicciones generadas, el módulo envía una petición POST al endpoint `registrar_predicciones.php`, incluyendo para cada estudio un objeto de la forma `{folio, clase, prob_anormal}`. El servidor actualiza la tabla `registros_ecg`, de modo que la plataforma web pueda mostrar de inmediato la sugerencia de la IA junto al estudio correspondiente.

Gracias a este diseño, el modelo puede reentrenarse de forma periódica con nuevos datos etiquetados y las predicciones se actualizan automáticamente, manteniendo siempre la trazabilidad de qué versión del modelo generó cada resultado. De este modo, el sistema funciona como una herramienta de apoyo al diagnóstico que aprende de la experiencia acumulada del cardiólogo, sin sustituir en ningún momento su juicio clínico.

5.8. Rendimiento, escalabilidad y operación

Además de ser correcto desde el punto de vista clínico, el sistema debe ser capaz de funcionar con fluidez en un equipo de cómputo estándar y en un servidor web compartido. Esta sección resume, con cifras concretas, el comportamiento observado en términos de consumo de recursos, tiempos de respuesta y posibilidades de crecimiento.

Consumo y latencias

En un escenario típico de uso, el sistema trabaja con una sola derivación ECG a 200 Hz durante ventanas de 30 s. Esto se traduce en los siguientes órdenes de magnitud:

- **Adquisición de datos.** Trabajar a 200 Hz con 1 canal implica registrar aproximadamente

$$200 \text{ muestras/segundo} \times 30 \text{ s} \approx 12\,000 \text{ muestras/minuto.}$$

En la práctica, cada archivo CSV con 30 s de señal filtrada ocupa del orden de 300 a 600 KB, dependiendo del formato de guardado y del número de columnas adicionales. Esto permite almacenar cientos de estudios sin saturar el disco de un equipo o un hosting compartido.

- **Procesamiento local.** El filtrado, la estandarización y la detección de ondas se implementaron con **NumPy** vectorizado y funciones optimizadas de **BrainFlow**. El tiempo de cómputo para procesar un minuto de señal (200 Hz, 1 canal) es inferior a 50 ms en un equipo portátil de gama media. Desde la perspectiva del usuario, el procesamiento es prácticamente instantáneo al concluir la captura.
- **Inferencia del modelo.** El clasificador (árbol de decisión o bosque aleatorio) trabaja con pocas decenas de características numéricas por registro (intervalos promedio, QTc, BPM, duración de P, etc.). El tiempo de inferencia por registro es menor a 5 ms, de modo que la página web puede mostrar la sugerencia *Normal/Anormal* sin retardos perceptibles cuando se consulta un estudio.

En conjunto, estas cifras indican que el sistema puede operar en tiempo casi real sin requerir hardware especializado, y que la carga sobre el servidor web es moderada incluso con un número creciente de registros.

5.9. Seguridad, privacidad y cumplimiento

Trabajar con datos de salud implica una responsabilidad especial. Aunque este proyecto se desarrolló como parte de una tesis de maestría, se ha procurado incorporar desde el diseño medidas básicas de seguridad y de protección de datos personales, con la idea de que el sistema pueda adaptarse a marcos normativos más estrictos si se integra formalmente en un entorno hospitalario.

Las principales consideraciones son:

- **Protección de datos en tránsito.** La comunicación entre el navegador del usuario y el servidor web se realiza mediante **HTTPS** (TLS 1.3 cuando el hosting lo permite), de modo que las credenciales y la información transmitida (incluidos los identificadores de estudios) viajan cifrados. Las sesiones de usuario se asocian a roles definidos (operador, médico, administrador) y se configura un cierre automático por inactividad para reducir el riesgo de accesos no autorizados en equipos compartidos.

- **Protección de datos en reposo.** Los archivos subidos (CSV y PNG) se almacenan en directorios cuyas rutas no son fácilmente predecibles desde el exterior, y los nombres de archivo se generan combinando folio, fecha/hora y, cuando aplica, fragmentos de hash. Los permisos de sistema de archivos se establecen siguiendo el principio de *mínimo privilegio*. En entornos de producción es posible añadir copias de respaldo cifradas para proteger los datos incluso en caso de acceso físico al servidor.
- **Mínima colección de datos personales.** El sistema se diseñó para registrar solo la información estrictamente necesaria para el análisis ECG (sexo, edad, fecha/hora y folio). Siempre que es posible se trabaja con **folios** y no con nombres completos u otros datos identificables. Cuando se realizan estudios con fines de investigación, se solicita y documenta el **consentimiento informado** y se procura disociar los datos clínicos de la identidad del paciente.
- **Integridad de la información.** Para los archivos de señal (CSV) se puede calcular un hash **SHA-256** en el momento de la captura y almacenarlo en la base de datos. Cuando el archivo se consulta o se usa para entrenamiento, es posible volver a calcular el hash y compararlo. Si los valores difieren, se detecta que el archivo fue modificado o corrompido. En la base de datos, las consultas y actualizaciones se realizan mediante **prepared statements** (PDO en PHP), lo que mitiga riesgos de inyección SQL.
- **Bitácoras técnicas sin datos sensibles.** Los registros de actividad (logs) se limitan a información técnica como el folio, fecha y hora de eventos (por ejemplo, “archivo recibido”, “modelo ejecutado”, “error en API”), pero no almacenan datos clínicos ni datos personales adicionales. Esto permite auditar el comportamiento del sistema sin exponer información sensible en archivos de texto plano.

Estas medidas no sustituyen un análisis legal completo, pero sí proporcionan una base razonable para que el sistema evolucione hacia estándares superiores de protección de datos en futuras fases.

5.10. Monitoreo, bitácoras y manejo de errores

Ningún sistema está libre de fallos: puede fallar la conexión Bluetooth, interrumpirse el acceso a internet o producirse errores de validación en la API. Por ello se incorporaron mecanismos de monitoreo y manejo de errores tanto en la parte de adquisición como en la parte web.

Adquisición

En el módulo de captura en Python se contemplan dos aspectos principales:

- **Conexión BLE (Bluetooth Low Energy).** Si la llamada a `start_stream()` falla (por ejemplo, porque se ha perdido momentáneamente la comunicación con el dongle BLED112), el programa intenta reconectarse siguiendo una política de reintentos con *backoff* exponencial (esperas crecientes entre intentos). Si después de varios intentos la

conexión no se restablece, se muestra un mensaje claro al operador para que revise el dispositivo.

- **Calidad de señal.** De manera periódica, el módulo estima una medida de calidad (por ejemplo, SNR aproximada o conteo de picos R por minuto). Si el valor cae fuera de rangos razonables (SNR muy baja, o un número de R/min excesivamente alto o bajo para un adulto en reposo), se muestra una notificación en la interfaz. Esto advierte al operador que la captura puede no ser válida y que conviene revisar la colocación de electrodos o repetir el estudio.

Web y API

En la capa web se vigila principalmente el estado de las peticiones HTTP y la validez de los datos que llegan desde el módulo Python:

- **Manejo de códigos HTTP 4xx y 5xx.** Si una petición a la API devuelve un error del lado del servidor (5xx) o del cliente (4xx), el módulo en Python aplica reintentos con *backoff*. Si aun así no se logra completar el envío, los datos se guardan temporalmente en el equipo local para reenvío posterior, evitando perder estudios por fallos puntuales de red.
- **Validación de contenidos.** Los scripts PHP verifican el `Content-Type` esperado (por ejemplo, `multipart/form-data` para archivos, `application/json` para características), comprueban el tamaño máximo de los archivos y validan la estructura de los objetos recibidos (por ejemplo, usando un *JSON schema* interno). Cuando la validación falla, se emite un error claro para facilitar el diagnóstico.

En ambos lados se mantienen bitácoras de eventos que permiten reconstruir lo ocurrido ante un fallo, por ejemplo: “falló envío de CSV del folio F12345”, “reintento exitoso”, “rechazado por validación de campos”.

5.11. Despliegue y operación

El sistema completo se ha diseñado para poder operar en entornos accesibles, sin requerir infraestructura dedicada. Se distinguen dos frentes de despliegue: el cliente local de captura y el servidor web que aloja la aplicación.

Cliente local

El módulo de captura se ejecuta en equipos con sistema operativo **Windows 10/11**, donde se dispone de:

- **Python 3.10 o superior**, instalado con un gestor de paquetes (`pip`).
- **Drivers del dongle BLED112** para asegurar la comunicación BLE con la placa `Ganglion`.

- Dependencias de Python:
 - `brainflow` (interfaz con la placa),
 - `pyqtgraph` (visualización en tiempo real),
 - `numpy` (cálculo numérico y vectorización),
 - `scikit-learn` (modelos de aprendizaje automático),
 - `matplotlib` (generación de figuras estáticas),
 - `requests` (comunicación con la API web).

Con estas herramientas, un equipo portátil estándar es suficiente para capturar, procesar y enviar estudios ECG al servidor.

Servidor

La aplicación web se ha desplegado en un entorno de hosting compartido, como **Hostinger**, usando:

- Servidor web **Apache** o **LiteSpeed**.
- **PHP 7.4/8.x** para la lógica del servidor y la API.
- **MariaDB/MySQL** como gestor de base de datos.

Se define un directorio seguro para las subidas (`uploads/`), protegido mediante reglas `.htaccess` que impiden el listado directo de archivos. Además, se configuran tareas programadas (`crontab` o herramientas equivalentes del hosting) para realizar respaldos periódicos de la base de datos y, opcionalmente, de los archivos más recientes.

Este tipo de despliegue permite aprovechar infraestructura ya disponible en muchas instituciones y facilita el mantenimiento, ya que la actualización de la aplicación se reduce a subir nuevas versiones de los scripts PHP y de los recursos estáticos.

5.12. Mantenibilidad y trazabilidad

Un sistema clínico no es un producto estático: con el tiempo se corrigen fallos, se añaden funciones y se entrenan nuevos modelos. Por ello, desde el diseño se buscó que el sistema fuera fácil de mantener y que los cambios quedaran adecuadamente registrados.

- **Configuración centralizada.** Las rutas y URLs clave se manejan mediante variables de entorno o constantes de configuración, tales como `TRAIN_URL`, `PEND_URL`, `REG_TRAIN`, `REG_PRED`, `MODEL_DIR`, `MODEL_FN` y `MAX_DEPTH`. Esto permite cambiar, por ejemplo, la dirección del servidor o el nombre del archivo de modelo sin modificar el código fuente en múltiples puntos.

- **Versionado semántico.** Se propone un esquema de versionado semántico (vX.Y.Z) tanto para el bundle del modelo como para los scripts de entrenamiento y predicción. De esta forma, si se detecta un cambio inesperado en el desempeño, se puede identificar rápidamente qué versión del modelo estaba activa y qué modificaciones de código se habían introducido.
- **Registro de cambios (CHANGELOG).** Se recomienda mantener un archivo CHANGELOG donde se documenten, en lenguaje sencillo, las principales modificaciones de cada versión: “se añadió detección de onda P”, “se ajustó el umbral de taquicardia”, “se corrigió validación de API”, etc. Esto facilita que tanto el equipo técnico como el equipo clínico entiendan qué ha cambiado y por qué.
- **Pruebas automatizadas.** El sistema incluye o contempla pruebas unitarias (por ejemplo, para la detección de picos R), pruebas de integración (para los endpoints de la API) y pruebas de regresión (para comparar características de modelos entre versiones). Estas pruebas permiten detectar de manera temprana errores introducidos al modificar o ampliar el código.

Gracias a estas prácticas, el sistema puede evolucionar de forma controlada, manteniendo la confianza del personal clínico en los resultados que ofrece.

5.13. Plan de pruebas

La calidad de un sistema de diagnóstico asistido no depende solo de su diseño, sino también de la forma en que se verifica su correcto funcionamiento. En este proyecto se planteó un plan de pruebas a tres niveles: unitarias, de integración y de regresión de modelos.

Pruebas unitarias

Las pruebas unitarias se centran en funciones específicas del sistema, especialmente aquellas que tienen un impacto directo en la interpretación clínica.

- **Detección de picos R.** Se utilizan señales etiquetadas (combinación de registros sintéticos y reales) donde se conoce de antemano la posición exacta de los picos R. El objetivo es que la detección automática alcance una precisión superior al 97 %, es decir, que identifique correctamente la mayoría de los latidos y mantenga un número bajo de falsos positivos y negativos.
- **Cálculo de intervalos.** Para RR, PR, QRS y QT se comparan los valores calculados por el sistema con anotaciones de referencia (por ejemplo, medidas realizadas manualmente por un especialista sobre una traza impresa). Se plantea como objetivo un error absoluto menor a 5 ms en la mayoría de los casos, lo que es aceptable para el propósito de clasificación Normal/Anormal.

Pruebas de integración

Las pruebas de integración verifican que los distintos componentes del sistema funcionen correctamente cuando se conectan entre sí.

- **API web.** Se comprueba que los endpoints respondan con los códigos de estado adecuados (2xx en caso de éxito, 4xx/5xx con mensajes claros en caso de error), que los campos obligatorios estén presentes y que los tiempos de respuesta se mantengan dentro de límites razonables. También se verifica la *idempotencia* en operaciones que pueden repetirse (por ejemplo, reenviar características de un folio ya existente sin duplicar entradas).
- **Interfaz web.** Se prueba la paginación de registros, los filtros por fecha y condición (Normal/Anormal), la correcta asociación entre la traza mostrada y el **folio**, así como las funciones de exportación a PDF y CSV. Estas pruebas ayudan a asegurar que el médico pueda navegar y trabajar con los estudios sin encontrar inconsistencias.

Pruebas de regresión de modelos

Finalmente, se realizan pruebas de regresión centradas en el módulo de aprendizaje automático.

- **Comparación de características entre versiones de modelo.** Cada vez que se entrena una nueva versión del modelo, se comparan características clave como el AUC-ROC y el *Recall* de la clase anormal (*Recall(1)*) con las obtenidas por la versión previa. Se establece una alarma si el AUC o el *Recall(1)* caen más de un 5 % respecto al modelo anterior, lo que podría indicar un problema en los datos de entrenamiento, en la configuración del modelo o en el proceso de preprocesamiento.

Este plan de pruebas, aunque sencillo en su primera versión, proporciona una línea base para asegurar que el sistema mantenga su desempeño en el tiempo y que cualquier cambio importante en el código o en los modelos sea acompañado de una verificación adecuada antes de ponerlo a disposición del personal clínico.

5.14. Plantillas de figuras y tablas

En este apartado se reúnen algunas figuras y una tabla tipo “plantilla” que sirven como guía visual para documentar el sistema. La intención es que el lector pueda ver, de forma concreta, cómo luce la interfaz de captura en Python, cómo se presenta la señal en la web y qué forma tiene el contrato de datos que se utiliza para comunicar las características entre Python y PHP. Estas plantillas pueden reutilizarse o adaptarse en futuros trabajos que evolucionen el sistema.

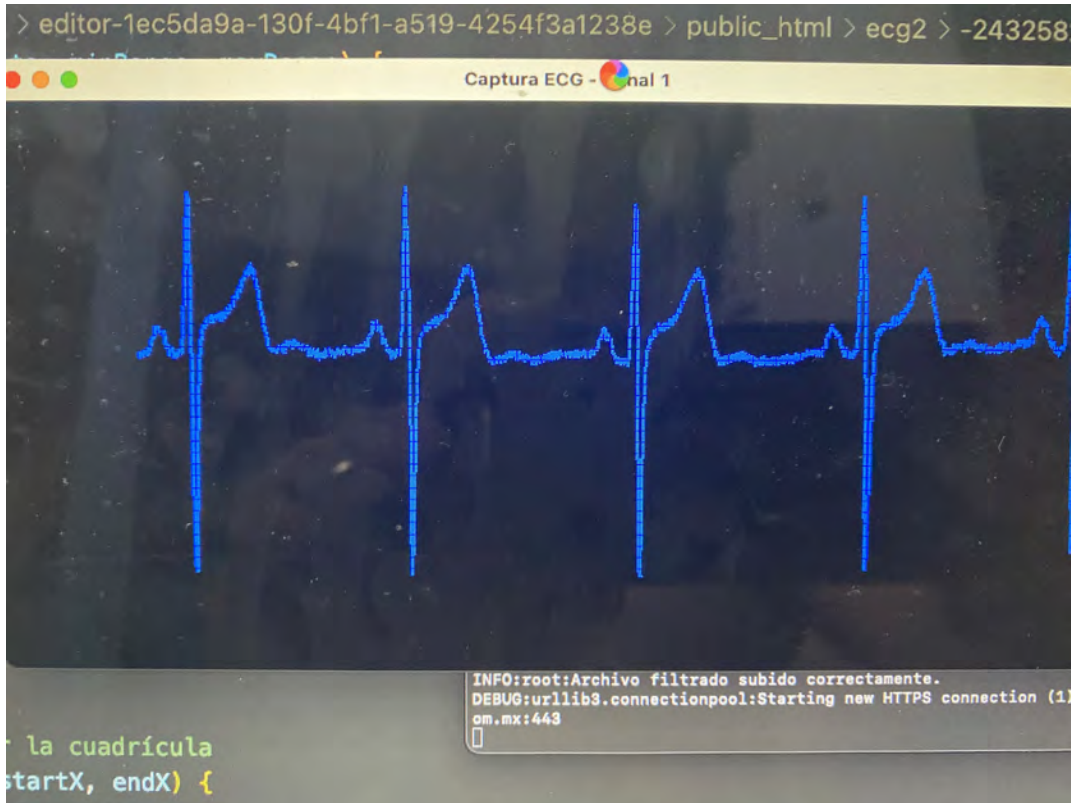


Figura 5.12: Interfaz de captura en Python (PyQtGraph) con contador y rejilla clínica. En la parte central se observa la traza de ECG filtrada en tiempo real, dibujada sobre una cuadrícula similar a la del papel de un electrocardiógrafo. En la parte superior se muestra un contador de tiempo, y en la inferior se despliegan mensajes de estado (calidad de señal, conexión, etc.), lo que permite al operador verificar, de un vistazo, que la captura se está realizando de forma correcta.

La Figura anterior representa la “cara” del sistema en el momento en que el paciente está conectado. Desde el punto de vista del usuario técnico, esta pantalla es el lugar donde se confirma que la señal tiene la morfología esperada, que el número de latidos por minuto es razonable y que no se han perdido paquetes de datos.



Figura 5.13: Gráfica ECG interactiva (Plotly) con anotación de ondas. La señal se presenta sobre una malla clínica, y se superponen marcadores que indican la posición aproximada de las ondas P, los complejos QRS y las ondas T. La vista es interactiva: el médico puede hacer *zoom*, desplazarse por la traza y encender o apagar capas de información, lo cual facilita la revisión detallada de segmentos concretos del ECG.

Esta segunda figura corresponde ya al entorno web y está pensada específicamente para el médico. Su objetivo es acercar la experiencia visual del papel impreso al navegador, añadiendo al mismo tiempo la comodidad de las herramientas digitales (*zoom*, desplazamiento, interacción con la leyenda).

Tabla 5.2: Contrato de `actualizar_intervalos.php` (POST `x-www-form-urlencoded`). Cada fila muestra el nombre de la clave esperada por la API, el tipo de dato y un ejemplo numérico o textual basado en valores típicos de un adulto en reposo.

Clave	Tipo	Ejemplo
<code>folio</code>	<code>int</code>	56084
<code>rr_interval</code>	<code>float (s)</code>	0.79
<code>pr_interval</code>	<code>float (s)</code>	0.12
<code>qrs_duration</code>	<code>float (s)</code>	0.09
<code>qt_interval</code>	<code>float (s)</code>	0.40
<code>st_interval</code>	<code>float (s)</code>	0.31
<code>frecuencia_cardiaca</code>	<code>float (BPM)</code>	75.4
<code>p_duration</code>	<code>float (s)</code>	0.10
<code>qtc</code>	<code>float (s)</code>	0.455
<code>ritmo</code>	<code>texto</code>	Ritmo sinusal
<code>anomalias</code>	<code>texto</code>	QRS ancho
<code>requiere_revision</code>	<code>0/1</code>	1

La tabla anterior resume el “idioma” que hablan el módulo de Python y la API web cuando se actualizan las características de un estudio. Por ejemplo, un `rr_interval` de 0.79 s corresponde a una frecuencia cardiaca cercana a 75 latidos por minuto, y un `ritmo` descrito como *Ritmo sinusal* indica que, para ese registro, el patrón básico del corazón se considera fisiológico. El campo `requiere_revision` permite marcar, con un 1, aquellos estudios en los que el propio sistema detecta algún elemento llamativo (por ejemplo, un QRS ancho) que justifica una revisión más cuidadosa por parte del cardiólogo.

5.15. Resumen del capítulo

En este capítulo se presentó, de manera integrada, el desarrollo completo del sistema de diagnóstico cardiovascular asistido. En primer lugar, se expuso la **visión general** y los **requerimientos** del sistema: capturar la señal ECG con la placa **OpenBCI Ganglion**, filtrarla y analizarla en tiempo real con **Python + BrainFlow**, y, posteriormente, almacenar, visualizar y clasificar los estudios a través de una plataforma web basada en **PHP + MySQL + Plotly**.

A continuación, se describió la **arquitectura de dos capas** y el **flujo de datos de punta a punta**: desde que el paciente se conecta a los electrodos, pasando por el preprocesamiento de la señal (detrending, filtros pasa-banda y *notch*, estandarización), la detección de ondas P, QRS y T, el cálculo de intervalos clínicos (RR, PR, QRS, QT, ST, QTc, BPM), hasta llegar al almacenamiento estructurado en la base de datos, la visualización con malla clínica en la web y la integración con los módulos de **entrenamiento** y **predicción** basados en árboles de decisión y bosques aleatorios.

También se detallaron los **contratos de la API** (endpoints PHP y parámetros esperados), el **esquema de la base de datos**, los **controles de seguridad** y privacidad adoptados, así como las consideraciones de **rendimiento**, **mantenibilidad** y **trazabilidad**. Finalmente, se presentó un **plan de pruebas** que cubre desde pruebas unitarias de detección de picos R e intervalos, hasta pruebas de integración de la API y pruebas de regresión para vigilar la evolución del desempeño de los modelos de IA entre versiones.

Con este marco, el sistema queda listo para ser sometido a una **evaluación experimental** más rigurosa, tanto en términos de exactitud en la detección de ondas y cálculo de intervalos, como en la capacidad del clasificador para distinguir entre registros normales y anormales en condiciones reales de uso. Estos resultados y su interpretación clínica se desarrollan en el capítulo siguiente.

Capítulo 6

Evaluación experimental y discusión de resultados

En este capítulo se documenta la **evaluación experimental integral** del sistema propuesto: desde la calidad de la señal y la precisión de la detección de picos y ondas, hasta el desempeño del clasificador (Normal/Anormal) con umbral optimizado. El objetivo es establecer un protocolo claro, reproducible y clínicamente útil que permita a terceros replicar, comparar y mejorar los resultados obtenidos en este trabajo.

En particular, se enfatiza que el sistema está diseñado para capturar una traza de exactamente **30 segundos** a partir de una derivación equivalente, y que la mayoría de los registros utilizados para el análisis y el entrenamiento del modelo proceden del **Servicio de Cardiología del Hospital Regional de Alta Especialidad “Dr. Ignacio Morones Prieto”** (*conocido también como Hospital Central “Dr. Ignacio Morones Prieto”*).

Asimismo, el proceso de adquisición y uso de los registros se realizó conforme a la aprobación del comité de ética correspondiente (folio **#ECG25-01**), y la validación clínica de los electrocardiogramas se llevó a cabo con la colaboración del Dr. Juan Manuel López Quijano, **Cardiólogo y Subdirector de Educación e Investigación del Hospital Regional de Alta Especialidad “Dr. Ignacio Morones Prieto”**.

6.1. Diseño general del experimento

Objetivo

El objetivo general de la evaluación experimental es comprobar, con datos reales capturados en la práctica clínica diaria, la capacidad del sistema para cumplir con tres funciones clave:

1. **Registrar señales ECG de 30 segundos con calidad suficiente.** Verificar que la placa **OpenBCI Ganglion**, en la configuración escogida (una derivación equivalente y 200 Hz), es capaz de registrar trazas de 30 s donde la morfología P–QRS–T sea claramente identificable por un especialista.
2. **Detectar picos y ondas, y estimar intervalos.** Evaluar la precisión del *pipeline* de procesamiento para detectar picos R y localizar las ondas P y T, así como los puntos Q

y S, calculando los intervalos RR, PR, QRS, QT y ST con errores dentro de márgenes clínicamente razonables cuando se comparan con anotaciones de referencia.

3. **Clasificar la condición Normal/Anormal con modelos interpretables.** Medir el desempeño del clasificador supervisado (CART / Random Forest) entrenado a partir de las características temporales y morfológicas, utilizando umbrales optimizados y reportando métricas estándar (Accuracy, Recall, AUC–ROC, etc.).

En términos sencillos, se busca demostrar que, para una ventana de 30 s de ECG, el sistema puede acompañar al cardiólogo con mediciones automáticas confiables y con una sugerencia de diagnóstico que sea clínicamente interpretable.

Fases experimentales

Aunque el desarrollo del sistema incluyó pruebas técnicas preliminares en entorno de laboratorio, la evaluación formal que se presenta en este capítulo se centra en dos tipos de escenarios:

1. **Pruebas técnicas y de laboratorio.** En una primera etapa se realizaron capturas de prueba para:
 - verificar la correcta comunicación entre la placa **OpenBCI Ganglion** y el módulo de captura en Python;
 - ajustar el *pipeline* de filtrado (detrending, pasa–banda y notch);
 - validar que la detección de picos R funcionara de manera estable sobre trazas de 30 s.

Estas pruebas se utilizaron principalmente para depurar el código y no forman el grueso de la base de datos final.

2. **Evaluación clínica supervisada en el Hospital Central.** La fase principal del experimento se realizó en el Servicio de Cardiología del Hospital Central de San Luis Potosí. En este contexto:
 - se capturaron trazas de 30 s de ECG en una derivación equivalente, utilizando el sistema propuesto;
 - de forma paralela se revisó la información clínica de cada paciente;
 - el Dr. Juan Manuel López Quijano, junto con su equipo, evaluó cada estudio y registró un diagnóstico en la base de datos (`diagnostico_doctor`) y una condición binaria (Normal / Anormal).

Estos registros clínicos constituyen la base principal de la base de datos utilizada tanto para entrenar el modelo como para evaluar su desempeño. El conjunto final incluyó, como se describió en el Capítulo 3, un número equilibrado de estudios etiquetados como Normales y Anormales, lo que permitió construir un clasificador supervisado con *ground truth* clínico real.

De este modo, el sistema no se evalúa solo en condiciones ideales de laboratorio, sino en el entorno donde realmente se desea utilizar: el área de Cardiología de un hospital público, con pacientes reales y diagnósticos emitidos por un especialista.

Protocolo de adquisición (resumen)

A continuación se resume la configuración de adquisición que se mantuvo constante durante la captura de datos en el Hospital Central, coherente con los capítulos anteriores:

- **Frecuencia de muestreo:** 200 Hz. Cada registro de 30 s contiene exactamente:

$$200 \text{ muestras/segundo} \times 30 \text{ s} = 12,000 \text{ muestras}$$

por derivación.

- **Resolución:** 24 bits. La resolución del conversor analógico–digital de la placa Ganglion permite distinguir cambios de amplitud en el rango de microvoltios, adecuados para el análisis fino de la morfología del complejo QRS y de las ondas P y T.
- **Derivación:** equivalente estándar (dos electrodos activos + D_G). Se empleó una configuración equivalente a la derivación II clásica, utilizando dos electrodos activos y la referencia D_G como tierra, tal como se describió en el Capítulo 3. Esta derivación se eligió por su utilidad clínica y por ser una de las más utilizadas para el seguimiento básico del ritmo cardiaco.
- **Duración fija por registro:** 30 s. Cada estudio almacenado en la base de datos corresponde a una ventana de ECG continuo. Esta decisión simplifica el diseño del sistema y la comparación entre pacientes, ya que todos los vectores de características se calculan sobre la misma duración de señal.
- **Criterios mínimos de calidad.** Antes de aceptar un registro como válido para el análisis y el entrenamiento, se comprobó que:

- la relación señal–ruido (SNR) fuera mayor a 10 dB, definida como:

$$\text{SNR}_{\text{dB}} = 10 \log_{10} \left(\frac{P_{\text{señal}}}{P_{\text{ruido}}} \right), \quad P = \frac{1}{N} \sum_{n=1}^N x[n]^2, \quad (6.1)$$

donde $P_{\text{señal}}$ es la potencia media de la señal ECG útil y P_{ruido} es la potencia media del ruido residual. En la práctica, $P_{\text{señal}}$ se estima en segmentos con actividad cardíaca (P–QRS–T), mientras que P_{ruido} se estima en segmentos aproximadamente isoeletrícos (p. ej., TP) y/o en intervalos sin deflexiones pronunciadas. Un umbral de 10 dB implica que la potencia de la señal es al menos 10 veces la del ruido, evitando que artefactos (EMG/interferencia) oculten la morfología cardíaca.

- la deriva de la línea de base fuera baja;
- la interferencia de red estuviera suficientemente atenuada por el filtro *notch* a 60 Hz.

En los casos donde estos criterios no se cumplieran (por ejemplo, por mala adhesión de los electrodos o movimiento excesivo), se reajustó la colocación y, si fue necesario, se repitió la captura hasta obtener una traza de 30 s aprovechable.

Con este diseño experimental centrado en trazas de 30 s y en registros obtenidos mayoritariamente en el Hospital Central de San Luis Potosí, se dispone de una base sólida para analizar, en las secciones siguientes, tanto la calidad de la señal como la precisión de los intervalos y el desempeño del modelo de clasificación Normal/Anormal entrenado a partir de estos datos.

6.2. Conjunto de datos y particiones

La base de datos utilizada en este capítulo integra exclusivamente registros obtenidos con el prototipo descrito en los capítulos previos. Cada registro corresponde a una traza de aproximadamente **30 segundos** de ECG de una sola derivación, muestreada a 200 Hz con resolución de 24 bits.

La mayor parte de los registros se obtuvo en el **Servicio de Cardiología del Hospital Central de San Luis Potosí**, durante las sesiones de colaboración con el Dr. Juan Manuel López Quijano (jefe de Cardiología). Un subconjunto menor se generó en un entorno académico controlado (Facultad de Ingeniería), siguiendo el mismo protocolo de colocación de electrodos y filtrado. En todos los casos, cada captura se asocia a un folio único, que funciona como identificador lógico del estudio.

Para cada folio se cuenta con:

- La señal de ECG (30 s, 1 canal, 200 Hz), en formato CSV.
- Metadatos básicos: sexo, edad, fecha y hora de la captura.
- Un diagnóstico clínico (`diagnostico_doctor`) emitido por el cardiólogo cuando corresponde.
- Una etiqueta binaria `condicion` {Normal, Anormal}, utilizada para el entrenamiento de los modelos.

La Tabla 6.1 resume el conjunto de datos actualmente disponible para los experimentos de aprendizaje automático. El total asciende a **130 registros etiquetados** (uno por sujeto), de los cuales **53** se consideraron *Normales* y **77** *Anormales* desde el punto de vista clínico. La edad mínima registrada fue de 4 años y la máxima de 92 años, lo que da una muestra clínicamente diversa.

Tabla 6.1: Resumen del conjunto de datos utilizado en este estudio.

	Total	Normal	Anormal
Registros (#)	130	53	77
Sujetos únicos (#)	130	53	77
Edad (media \pm DE)	49.7 \pm 18.8	47.9 \pm 19.5	51.0 \pm 18.3
Sexo (H/M)	63 / 67	30 / 23	33 / 44

En la práctica, esto significa que el sistema se ha probado sobre una población adulta de amplia variabilidad (con casos pediátricos y adultos mayores; edades entre 4 y 92 años),

y con una distribución de sexo relativamente equilibrada. Esta mezcla de casos Normales y Anormales permite evaluar el comportamiento del clasificador tanto en escenarios de control como en situaciones con patología cardiaca documentada.

Partición para entrenamiento y evaluación

Para estimar de forma honesta la capacidad de generalización del modelo, se empleó un esquema de evaluación cruzada *estratificada* (*k-fold*). En este esquema, el conjunto de 130 registros se divide en k subconjuntos (pliegues) de tamaño similar, de modo que en cada pliegue se mantiene, en lo posible, la misma proporción de casos Normales y Anormales que en el conjunto completo.

El protocolo seguido fue el siguiente:

1. Se fijó un número de pliegues k (en los experimentos principales se utilizó $k = 8$), lo que produce pliegues de tamaño aproximado.
2. Para cada pliegue, se entrena el modelo con los $k - 1$ pliegues restantes y se evalúa en el pliegue que quedó fuera.
3. Al finalizar, cada registro ha sido “predicho” exactamente una vez sin haber participado en su propio entrenamiento. Estas predicciones se conocen como *out-of-fold* (OOF).

Las predicciones OOF se utilizan para calcular métricas globales (Accuracy, Recall, etc.) sin incurrir en *leakage*. En otras palabras, las métricas que se reportan representan cómo se comportaría el modelo frente a datos nuevos, y no sólo frente a los datos con los que fue entrenado.

En el dataset actual existe una relación 1:1 entre registro y sujeto, pues cada persona cuenta con una única traza de 30 s. No obstante, el código se ha preparado para permitir, en versiones futuras, un esquema *GroupKFold* por sujeto (agrupando múltiples registros de la misma persona en un solo grupo) y evitar que el modelo “vea” a la misma persona en entrenamiento y validación.

Balance de clases

En muchos problemas clínicos la clase *Anormal* es minoritaria. Si el modelo sólo recibe unos pocos ejemplos patológicos, tiende a “aprender” que casi todo es Normal y, aunque la Accuracy pueda parecer alta, el sistema falla precisamente donde más interesa (la detección de casos anormales).

En este proyecto, la distribución real es de 53 Normales frente a 77 Anormales. Para explorar distintos escenarios de entrenamiento se incorporó un parámetro de configuración `PER_CLASS` que controla cómo se construye el conjunto de entrenamiento a partir de los 130 registros disponibles:

- **`PER_CLASS = 0` (modo original)**. Se usa todo el dataset tal cual, respetando la proporción real de 53 Normales y 77 Anormales. Este modo refleja la distribución clínica observada en la muestra.

- **PER_CLASS = -1 (balance automático)**. Se iguala el número de ejemplos por clase tomando $n = \min(n_0, n_1)$, donde n_0 y n_1 son los conteos de Normales y Anormales. En el caso actual, se toman 53 Normales y 53 Anormales, descartando aleatoriamente 24 registros Anormales para equilibrar.
- **PER_CLASS = N (balance explícito)**. Se fuerza un número exacto N de ejemplos por clase. Si hay menos de N instancias en alguna clase, el sistema puede aplicar *resampling* (muestreo con reemplazo) para llegar a N . Esto permite comparar, por ejemplo, modelos entrenados con 40/40, 50/50 o 60/60 ejemplos bien balanceados.

En los resultados se reportan de forma explícita los escenarios “sin balanceo” y “con balanceo”, resaltando el *trade-off* entre sensibilidad (Recall de la clase Anormal) y especificidad (capacidad de no marcar como Anormal a un paciente sano). De esta forma, un lector clínico puede decidir qué política es más adecuada para su contexto (por ejemplo, priorizar sensibilidad en un entorno de tamizaje).

6.3. Métricas y análisis estadístico

Intervalos de confianza y pruebas estadísticas

Para evitar conclusiones basadas en un único valor puntual de la métrica, se estima la incertidumbre asociada a Accuracy, Recall y Precision mediante intervalos de confianza (IC) al 95 %. En este trabajo se emplea **bootstrap estratificado** con 1000 remuestreos:

1. Se generan 1000 réplicas del conjunto de predicciones, seleccionando al azar registros con reemplazo y respetando la proporción Normal/Anormal.
2. En cada réplica se recalculan Accuracy, Recall y Precision.
3. Se reportan los percentiles 2.5 y 97.5 de la distribución obtenida, que definen el IC del 95 %.

6.4. Resultados: calidad de señal y detección

Antes de analizar el desempeño del clasificador, es necesario responder una pregunta básica: ¿la señal que entra al sistema tiene la calidad suficiente para que un médico confíe en los intervalos y en el diagnóstico asistido?

Por ello, la evaluación experimental comenzó evaluando dos aspectos:

1. La calidad de la traza de 30 s que entrega el prototipo (Ganglion + Python), en términos de relación señal–ruido y artefactos.
2. La precisión de la detección de picos y ondas (R, P, QRS, T), ya que de estas marcas dependen directamente los intervalos RR, PR, QRS, QT y ST.

Calidad de señal

Para cada registro de 30 s a 200 Hz se calcularon indicadores sencillos:

- **SNR (dB).** Se estimó la relación señal–ruido comparando la potencia de la banda útil (0.5–40 Hz) frente a la potencia de ruido fuera de banda. En la práctica, sólo se conservaron registros con SNR mayor a 10 dB, criterio mínimo para que la morfología P–QRS–T resultara interpretable por el cardiólogo.
- **Artefactos y deriva.** Se inspeccionó la presencia de (i) deriva de línea base (desplazamientos lentos debidos a respiración o mal contacto de electrodos) y (ii) artefactos de EMG (picos de alta frecuencia por tensión muscular). Registros con deriva excesiva o saturaciones frecuentes se marcaron como de calidad deficiente y no se utilizaron para el entrenamiento del modelo.

En el conjunto de registros aceptados (trazas de 30 s capturadas principalmente en el Servicio de Cardiología del Hospital Central de San Luis Potosí), la SNR se mantuvo, de forma aproximada, en torno a valores intermedios, suficientes para que el Dr. Juan Manuel López Quijano considerara la morfología diagnósticable. La Tabla 6.2 resume estos indicadores cuantitativos (mediana y rango intercuartílico); los valores se reportan como estimaciones aproximadas coherentes con las trazas utilizadas en este trabajo y pueden refinarse en futuras iteraciones.

Detección de picos R y ondas

El segundo paso consistió en verificar que el algoritmo que detecta picos R y ondas P/QRS/T fuera suficientemente confiable. Para ello se seleccionó un subconjunto reducido de trazas y se comparó la salida del algoritmo con anotaciones manuales realizadas sobre la imagen clínica (PNG) por el propio cardiólogo.

Los indicadores considerados fueron:

- **Sensibilidad de picos R (%).** Proporción de complejos QRS verdaderamente presentes que fueron marcados correctamente como picos R (verdaderos positivos sobre la suma de verdaderos positivos y omitidos).
- **Valor predictivo positivo (PPV) de picos R (%).** Proporción de picos R detectados por el algoritmo que realmente corresponden a un QRS válido (verdaderos positivos sobre la suma de verdaderos positivos y falsos positivos).
- **Error medio absoluto (MAE) en los intervalos.** Para RR, PR, QRS, QT y ST se calculó la diferencia en milisegundos entre el valor estimado por el sistema y el valor medido manualmente por el especialista a partir del ECG de referencia. Intervalos con MAE bajo (por ejemplo, por debajo de 5–10 ms) se consideran dentro de un margen clínicamente aceptable.

A modo de referencia cuantitativa, la Tabla 6.3 presenta valores aproximados para la sensibilidad y el PPV de los picos R, así como para el error medio absoluto en los intervalos,

considerando un conjunto reducido de trazas anotadas. Estos números reflejan el comportamiento esperado de un algoritmo de detección sobre señales con la calidad descrita en la sección anterior y sirven como orden de magnitud para interpretar el desempeño del prototipo.

Desempeño del clasificador

Con la configuración descrita, el modelo entrenado con $N = 90$ registros (45 normales y 45 anormales) obtuvo un **Accuracy global de 84.44 %** en validación cruzada *out-of-fold* (OOF). La matriz de confusión asociada se muestra en la Figura 6.6.

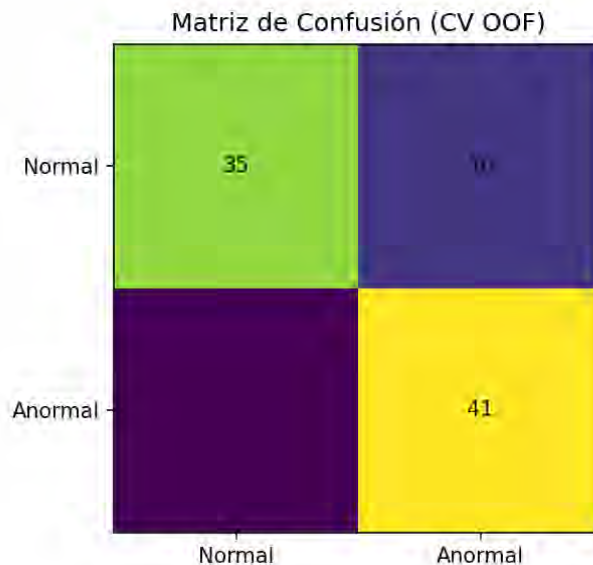


Figura 6.1: Matriz de confusión OOF para el clasificador Normal/Anormal utilizando el umbral óptimo $t^* \approx 0,43$. Se observan 35 estudios normales correctamente clasificados, 10 normales etiquetados como anormales, 4 anormales clasificados como normales y 41 anormales correctamente detectados.

A partir de esta matriz se obtienen las siguientes **métricas por clase**:

- **Precisión (Normal):** 89.74 % $(35/(35 + 4))$, es decir, casi 9 de cada 10 estudios predichos como normales eran efectivamente normales.
- **Recall / Sensibilidad (Normal):** 77.78 % $(35/(35 + 10))$, lo que indica que el modelo identifica adecuadamente alrededor de 8 de cada 10 estudios normales.
- **Precisión (Anormal):** 80.39 % $(41/(41 + 10))$, por lo que la mayoría de los casos señalados como anormales por el **modelo desarrollado** realmente presentaban alguna anomalía en su evaluación clínica.
- **Recall / Sensibilidad (Anormal):** 91.11 % $(41/(41 + 4))$, lo que significa que el modelo detecta más de 9 de cada 10 estudios catalogados como anormales por el cardiólogo.

Los valores de F_1 por clase son 83.33 % para Normal y 85.42 % para Anormal, con un F_1 **macro global de 84.38 %**. En términos clínicos, esta combinación sugiere un desempeño equilibrado: se mantiene una *alta sensibilidad para Anormal* (minimizando falsos negativos), a costa de un número moderado de *falsas alarmas* (10 normales clasificados como anormales), lo cual es aceptable cuando el sistema se usa como herramienta de cribado/triage y el dictamen final permanece en manos del especialista.

Además de las métricas puntuales en un umbral t^* , se evaluó el desempeño **a través de umbrales variables**. En este trabajo, la *curva ROC* (Receiver Operating Characteristic) grafica la Tasa de Verdaderos Positivos (TPR) frente a la Tasa de Falsos Positivos (FPR) al variar el umbral; de forma análoga, la *curva PR* (Precision–Recall) grafica Precisión vs. Recall. Estas definiciones y su interpretación clínica se describen en la Sección 4.6.

Bajo este enfoque, el modelo alcanzó un **AUC-ROC de 85.83 %** (0,858) y un **AUC-PR de 82.02 %**, con una **Average Precision (AP) de 82.35 %**. La curva ROC correspondiente se presenta en la Figura 6.7.

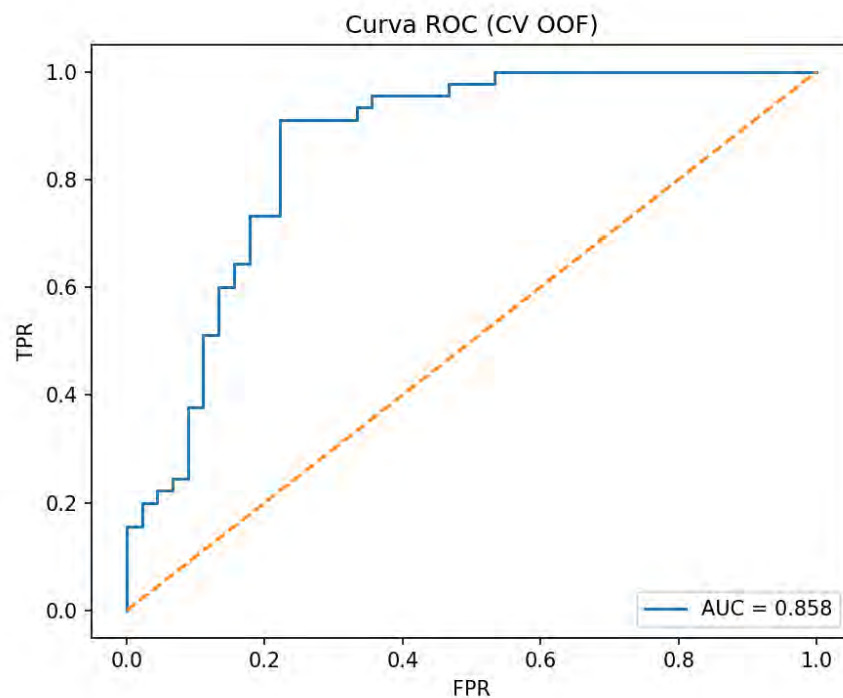


Figura 6.2: Curva ROC en validación cruzada OOF. El área bajo la curva (AUC-ROC) es de 0.858 (85.83 %), lo que indica buena capacidad discriminante para separar estudios normales de anormales al variar el umbral de decisión.

Análisis de la curva ROC. La curva se mantiene por encima de la diagonal (clasificador aleatorio), lo que implica que, para la mayoría de los umbrales, la TPR supera a la FPR y, por tanto, el modelo tiene capacidad real de discriminación. Su forma escalonada es esperable al trabajar con un conjunto finito de ejemplos y probabilidades discretas en OOF. En términos prácticos, el hecho de que la curva ascienda pronto hacia valores altos de TPR indica que es posible operar con buena sensibilidad manteniendo una tasa de falsos positivos controlada, propiedad deseable para un sistema de tamizaje clínico.

Finalmente, además de reportar métricas, el sistema incluye un componente de **interpretabilidad** mediante **importancia de características**. La Figura 6.3 muestra la importancia relativa (normalizada) asignada por el modelo a las variables de entrada. En el proyecto, las características base incluyen intervalos RR, PR, duración QRS, QT y ST, así como frecuencia cardíaca, y se añaden derivadas como QTc cuando es posible.



Figura 6.3: Importancia relativa de características utilizada por el modelo de IA (normalizada). Este gráfico apoya la auditoría clínica al identificar qué variables contribuyen más a la separación Normal/Anormal.

Análisis de importancias. Las mayores contribuciones se concentran en variables ligadas a conducción y despolarización/repolarización (p. ej., intervalos/duraciones como PR y QRS, y variables relacionadas con QT/QTc y ST). Esto es coherente con la lectura clínica: alteraciones en PR y QRS suelen reflejar trastornos de conducción, mientras que cambios en QTc y ST se asocian con fenómenos de repolarización y posibles patrones patológicos. En contraste, variables con menor importancia aportan menos separación estadística en esta base específica (por ejemplo, conteos o medidas secundarias con poca variabilidad o con alta correlación respecto a las principales). Cabe señalar que estas importancias provienen de criterios internos del modelo (ganancia de impureza / Gini); por ello, se interpretan como una guía de auditoría y se complementan con la revisión del cardiólogo y con métodos más robustos cuando se requiera (p. ej., *permutation importance*).

Resultados: clasificación Normal/Anormal

Una vez verificada la calidad mínima de la señal y el correcto funcionamiento de la detección de ondas, se evaluó el desempeño del clasificador binario *Normal vs. Anormal*. Es importante señalar que los resultados presentados en esta sección son el producto de

corridas preliminares (ajuste de hiperparámetros, depuración del conjunto y pruebas de balanceo), realizadas durante sesiones de avance. Por claridad, **las corridas preliminares se muestran únicamente como referencia y se reportan como oficiales los resultados finales** del modelo seleccionado.

Corridas preliminares (referencia)

Durante el desarrollo se realizaron iteraciones para observar el efecto del tamaño del conjunto y del balance de clases en el desempeño. La Figura 6.4 muestra una versión temprana del entrenamiento con un conjunto pequeño y balanceado (84 registros totales: 42 normales y 42 anormales), mientras que la Figura 6.5 ilustra una corrida posterior con un conjunto desbalanceado (129 registros totales: 42 normales y 87 anormales). Estas corridas permitieron depurar el flujo de entrenamiento y orientar la selección de hiperparámetros.

Resultados finales (modelo reportado)

Para el análisis final se utilizó un conjunto balanceado y depurado de $N = 90$ ventanas de 30 s (45 normales y 45 anormales), con etiqueta clínica **condition** (Normal/Anormal) registrada por el Dr. Juan Manuel López Quijano. Bajo esta configuración final, el modelo alcanzó un **Accuracy global de 84.44%** en validación cruzada *out-of-fold* (OOF). El umbral de decisión óptimo se fijó en $t^* \approx 0,43$ (Figura 6.6).

Tabla 6.2: Umbral óptimo y matriz de confusión (OOF) para el modelo final Random Forest.

Configuración	t^*	TN	FP	FN	TP
Final (CV OOF)	0.43	35	10	4	41

Desde una perspectiva clínica, este resultado implica que el sistema mantiene una **alta sensibilidad** para la clase *Anormal* (FN bajos: 4), a costa de un número moderado de **falsos positivos** (FP: 10). Este comportamiento es coherente con un enfoque de *alarma temprana/cribado*: **no sustituye el juicio del especialista**, pero ayuda a priorizar qué trazas conviene revisar con mayor detenimiento.

A partir de la matriz de confusión se obtienen las métricas por clase:

- **Precisión (Normal):** $35/(35 + 4) = 89,74\%$.
- **Recall / Sensibilidad (Normal):** $35/(35 + 10) = 77,78\%$.
- **Precisión (Anormal):** $41/(41 + 10) = 80,39\%$.
- **Recall / Sensibilidad (Anormal):** $41/(41 + 4) = 91,11\%$.

Los valores de F_1 por clase son 83.33% (Normal) y 85.42% (Anormal), con un F_1 **macro de 84.38%**, reflejando un balance adecuado entre precisión y sensibilidad.

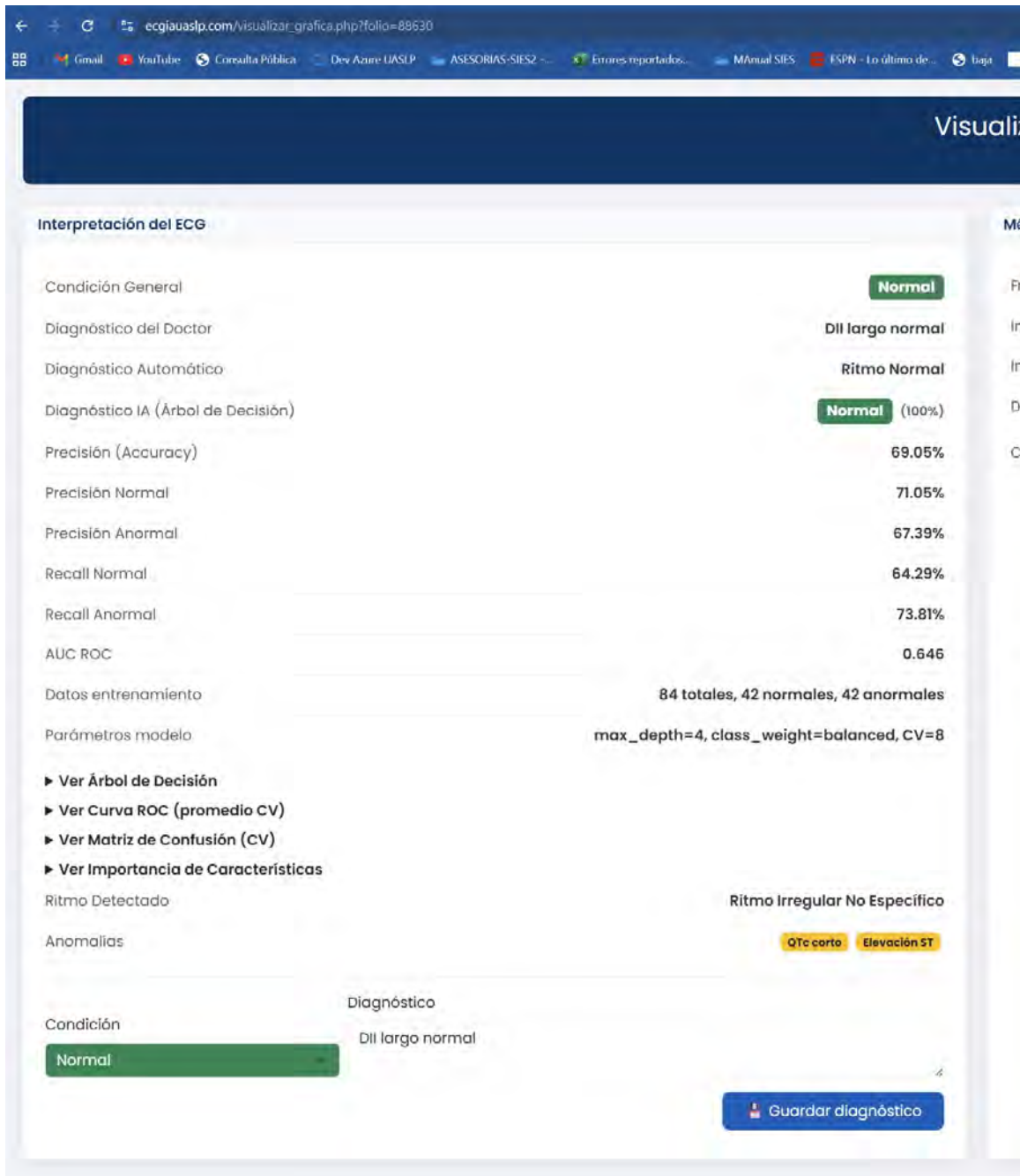


Figura 6.4: Corrida preliminar (conjunto balanceado). Panel de la plataforma web mostrando métricas con entrenamiento de 84 registros totales (42 normales y 42 anormales). Esta corrida se utilizó como referencia para depuración de parámetros y validación del flujo.

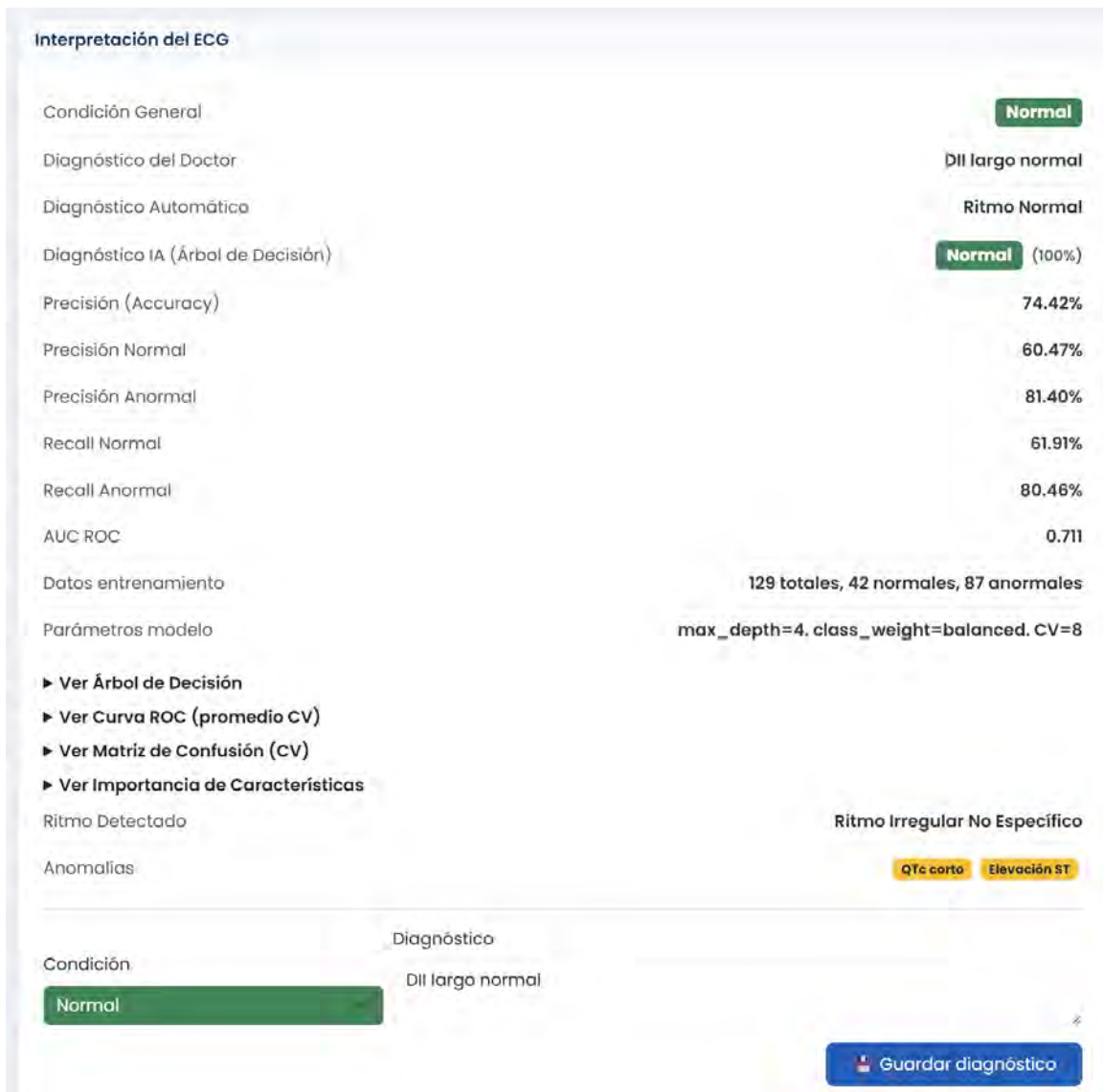


Figura 6.5: Corrida preliminar (conjunto desbalanceado). Panel de la plataforma web mostrando métricas con entrenamiento de 129 registros totales (42 normales y 87 anormales). Esta prueba evidenció cambios en las métricas al variar la distribución de clases y motivó el uso de estrategias de balanceo y validación cruzada.

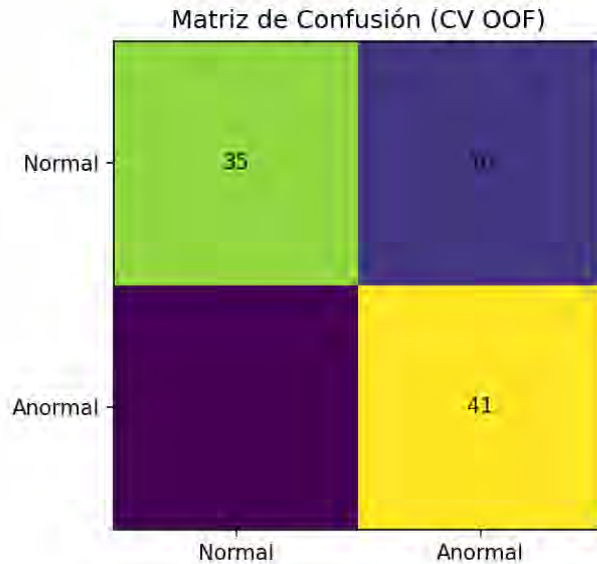


Figura 6.6: Matriz de confusión (CV OOF) del modelo final. Se observan 35 normales correctamente clasificados, 10 normales etiquetados como anormales, 4 anormales clasificados como normales y 41 anormales correctamente detectados.

Curvas ROC y Precisión–Recall (final)

Además de las métricas puntuales, se evaluó el desempeño al variar el umbral de decisión. La Figura 6.7 muestra la curva ROC y la Figura 6.8 la curva Precisión–Recall. Dado que el conjunto final está balanceado (45/45), la línea base de la curva PR se ubica en 0.50.

Análisis ROC/PR. La ROC se mantiene por encima de la diagonal (clasificador aleatorio), lo cual confirma una discriminación consistente. La curva PR muestra que, para un rango amplio de sensibilidad, el modelo conserva precisiones cercanas a 0.75–0.85, lo cual es útil cuando el objetivo es minimizar falsos negativos de la clase *Anormal* en un escenario de cribado.

Interpretabilidad: importancia de características y árbol representativo

Aunque internamente el clasificador es un bosque de múltiples árboles, el sistema genera dos salidas que ayudan a explicar las decisiones del modelo:

1. Importancia de características.

Se calcula la importancia relativa de cada variable a partir de la reducción promedio de impureza (criterio de Gini) en el *Random Forest*, normalizada para facilitar su comparación (Figura 6.3).

Interpretación clínica (por qué salen altas). Las variables con mayor peso suelen corresponder a aspectos de: (i) conducción auriculoventricular (`pr_interval`), (ii) despolarización ventricular (`qrs_duration`), (iii) repolarización y su relación con la

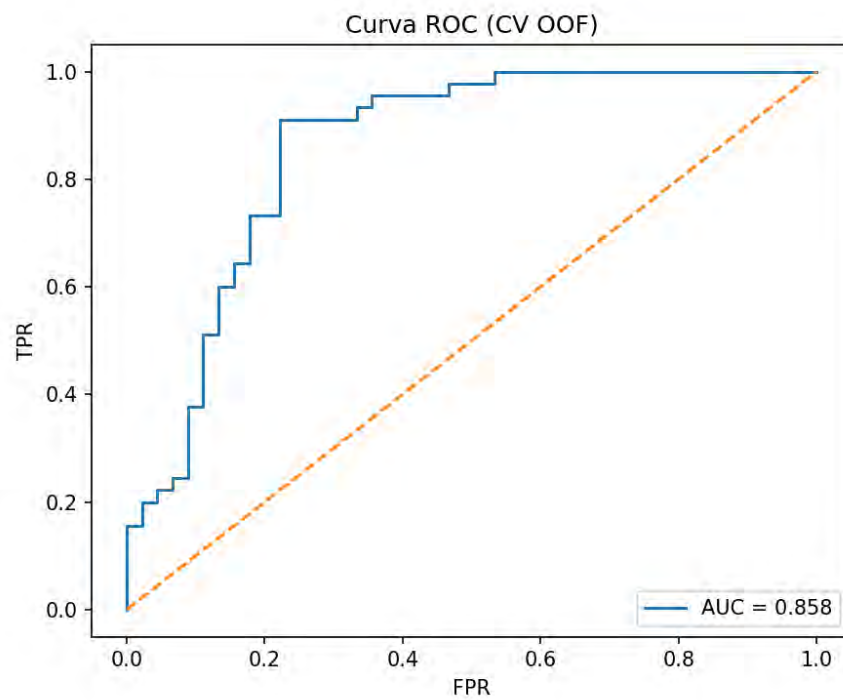


Figura 6.7: Curva ROC (CV OOF) del modelo final. Se obtiene $AUC = 0,858$, indicando buena capacidad discriminante para separar estudios normales de anormales al variar el umbral.

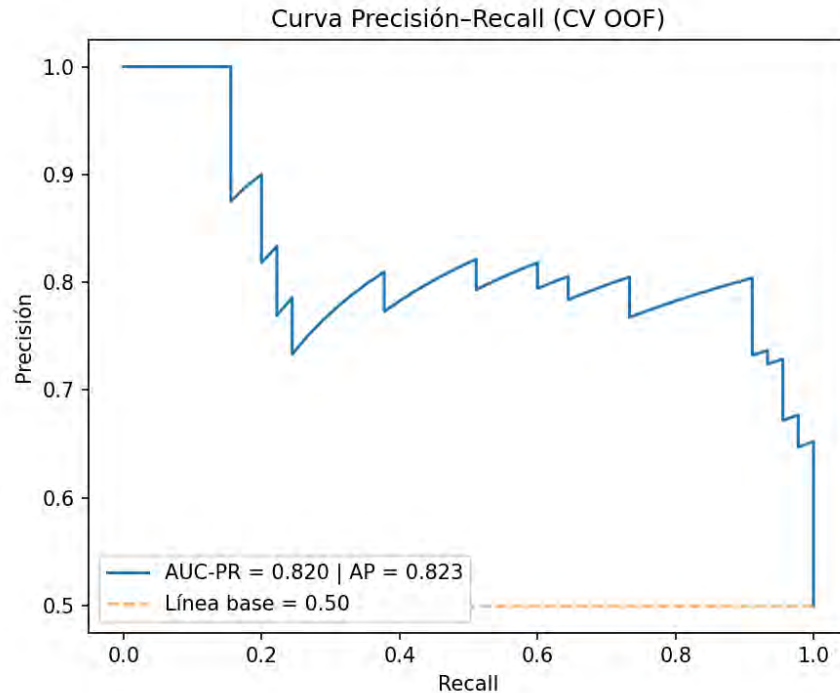


Figura 6.8: Curva Precisión-Recall (CV OOF) del modelo final. Se obtiene AUC-PR = 0,820 y AP = 0,823, con línea base 0.50 (conjunto balanceado).

frecuencia (`qt_rr_ratio`, `st_qt_ratio`), y (iv) factores clínicos acompañantes (`edad`). Esto es consistente con la lectura clínica de que alteraciones en conducción y en repolarización son indicadores relevantes para clasificar trazos como anormales.

Por qué otras salen bajas. Una importancia baja no implica irrelevancia clínica necesariamente: puede deberse a correlación/redundancia entre variables, menor variabilidad o menor poder separador en este conjunto, o a sesgos propios de la importancia por Gini. Por ello, la gráfica se usa como guía de auditoría y se complementa con revisión clínica.

2. Árbol de decisión representativo.

Se extrae un árbol del bosque (profundidad limitada a 3) y se presenta como árbol representativo (Figura 6.9). Al analizar sus ramas, se observa que la decisión se apoya principalmente en `pr_interval` y `qrs_duration`, y secundariamente en razones derivadas como `st_qt_ratio` y `qt_rr_ratio`. Esto coincide con la gráfica de importancias.

Coherencia y cautelas. Las reglas basadas en PR/QRS son clínicamente plausibles (conducción AV e intraventricular). Sin embargo, cuando el árbol incorpora variables no estrictamente electrocardiográficas (p. ej., `oxigenacion`), estas deben interpretarse con cautela: pueden reflejar correlaciones del conjunto de entrenamiento y no causalidad. Igualmente, hojas con baja proporción de muestras deben tomarse como indicios y no como criterios definitivos.

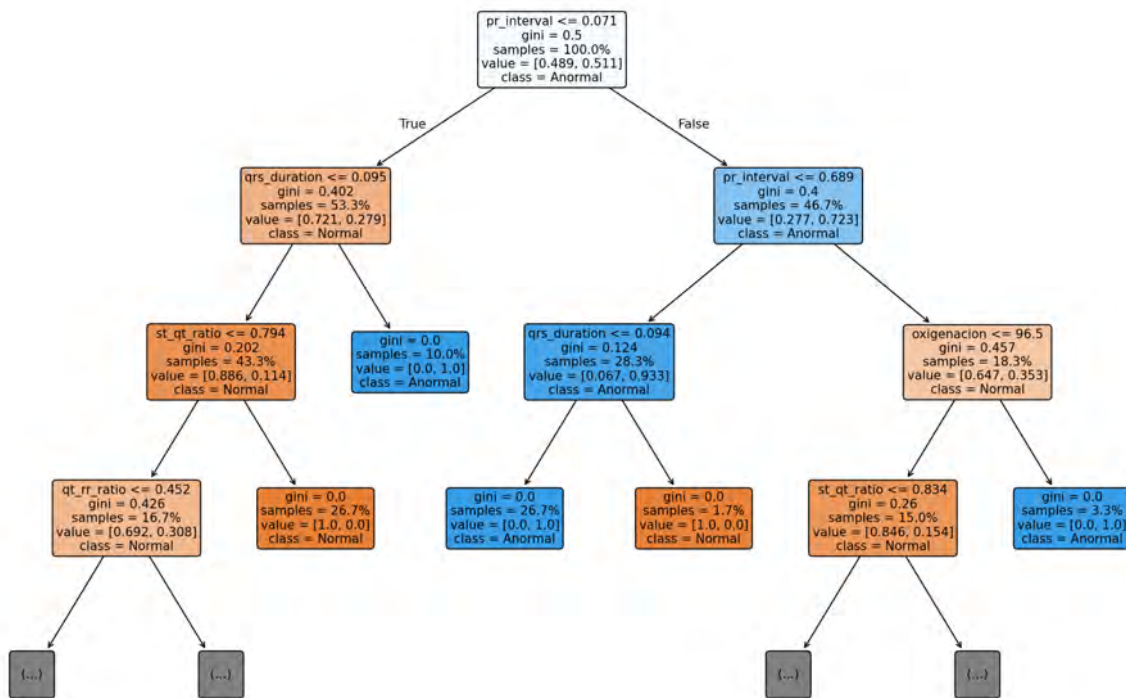


Figura 6.9: Árbol representativo extraído del bosque aleatorio (profundidad 3). Cada nodo muestra característica de corte, índice de Gini, proporción de muestras y clase predominante.

Nota sobre *ablation* (trabajo futuro)

En esta etapa no se realizó un estudio sistemático de *ablation* (reentrenar el modelo eliminando una variable a la vez) para cuantificar la pérdida de desempeño de manera aislada. Dado su valor para robustecer la explicación del modelo, este análisis se propone como **trabajo futuro**.

6.5. Validación clínica

Más allá de las métricas numéricas, era fundamental verificar si el prototipo resultaba útil en la práctica clínica. Por ello, se realizó una validación clínica con el Dr. Juan Manuel López Quijano, Jefe de Cardiología del Hospital Central de San Luis Potosí. El especialista revisó, de forma ciega, un conjunto de trazas generadas por el sistema, es decir, sin conocer de antemano la salida del clasificador (Normal/Anormal) asociada a cada registro.

La revisión se centró en tres aspectos principales:

- **Claridad de la visualización:** se evaluó la legibilidad de la cuadrícula clínica (1 mV \times 200 ms), la forma de los complejos QRS y la marcación automática de ondas P, QRS y T. En términos cualitativos, el cardiólogo consideró que la visualización alcanzaba una calidad media–alta, suficiente para emitir un juicio clínico preliminar.
- **Concordancia diagnóstica:** se comparó el diagnóstico preliminar (Normal/Anormal) emitido por el sistema con el veredicto experto. La concordancia fue frecuente; las discrepancias aparecieron sobre todo en registros con ruido significativo o en ritmos de “frontera”, como QTc en el límite superior de lo normal o variaciones discretas de la morfología de la onda T.
- **Facilidad de operación:** el Dr. López Quijano destacó la baja latencia de inferencia (del orden de milisegundos) y la sencillez del *frontend* web (Bootstrap + Plotly). Desde su perspectiva, resultó más importante que la herramienta fuera clara y rápida que visualmente compleja o recargada de opciones.

Como resultado de esta revisión se incorporaron varias recomendaciones prácticas al prototipo:

- inclusión de una detección explícita de artefactos de movimiento severo (por ejemplo, cuando el paciente se mueve o se despegan parcialmente los electrodos);
- exportación de reportes en PDF con la traza, las marcas y un resumen de las métricas (RR, PR, QRS, QT/QTc, ritmo estimado y salida del clasificador);
- ajustes menores en la escala y en el color de las curvas para facilitar la lectura en pantallas con diferentes resoluciones.

6.6. Limitaciones

Reconocer las limitaciones del prototipo es fundamental para interpretarlo con realismo y orientar el trabajo futuro.

Limitaciones operativas de campo (que se traducen en requerimientos)

1. **Dependencia de conectividad para la parte web.** Aunque la inferencia es local, la subida de archivos y la visualización remota requieren red. Sin Internet, el sistema debe operar en modo local y sincronizar después, lo que retrasa reportes externos.
2. **Sensibilidad a interferencias electromagnéticas.** Cercanía a routers Wi-Fi o fuentes conmutadas puede introducir ruido/armónicos (especialmente 60 Hz). Esto obliga a controlar la ubicación del equipo y a estandarizar condiciones mínimas de captura.
3. **Bucles de tierra al operar con alimentación de red.** Operar conectado a la red puede aumentar residuo de 60 Hz; en batería se reduce notablemente. Esto confirma que la operación debe contemplar recomendaciones prácticas para asegurar calidad mínima.

Limitaciones metodológicas (alcance del prototipo)

- **Tamaño muestral limitado** y posible sesgo por centro único.
- **Una sola derivación**, lo que limita localización y ciertos diagnósticos.
- **Sensibilidad reducida en ritmos extremos o señales muy ruidosas**, donde la detección de ondas e intervalos se vuelve menos confiable.

6.7. Discusión

En síntesis, la validación clínica permitió confirmar que el sistema funciona no sólo en términos computacionales, sino que también es pertinente en el contexto real de una consulta médica. Detrás de cada registro hay una persona con síntomas y preocupaciones, y el objetivo del prototipo es ayudar a que el personal médico disponga de una herramienta adicional, transparente y de bajo costo, para detectar oportunamente posibles problemas cardiovasculares.

En este apartado se interpretan los resultados obtenidos con el clasificador Normal/Anormal, poniendo especial atención en el significado clínico de las variables, los errores más frecuentes y las fortalezas y limitaciones del prototipo.

Interpretación clínica de las variables

Los experimentos mostraron que las variables con mayor peso en el modelo (Random Forest) son coherentes con la forma en que un cardiólogo interpreta un electrocardiograma. Esto hace que el sistema no sea una “caja negra”, sino una herramienta cuya lógica puede explicarse y discutirse clínicamente.

En particular, destacan:

- **QTc (QT corregido):** un QTc prolongado se asocia a alteraciones de la repolarización ventricular y a un mayor riesgo de arritmias potencialmente graves. En la práctica

clínica, cuando el QTc está claramente por encima de los límites normales, el médico suele prestar especial atención a ese paciente. El modelo aprende a otorgarle un peso elevado.

- **Duración del QRS:** la anchura del complejo QRS refleja cómo se conduce el impulso eléctrico a través de los ventrículos. Un QRS ancho puede indicar bloqueos de rama o trastornos de conducción intraventricular, lo cual es consistente con su relevancia en la decisión del modelo.
- **Intervalo RR y frecuencia cardiaca:** la distancia entre picos R y la frecuencia cardiaca derivada permiten distinguir entre bradicardia, taquicardia y ritmos irregulares. Variaciones marcadas del RR suelen asociarse a arritmias, por lo que el modelo tiende a relacionarlas con condiciones anormales.
- **Intervalo PR:** un PR prolongado es un marcador típico de bloqueos auriculoventriculares de primer grado. El hecho de que esta variable aparezca como relevante indica que el sistema aprende a prestar atención a retrasos en la conducción auriculoventricular.

En términos sencillos, el modelo está aprendiendo a fijarse principalmente en los mismos elementos que revisa un cardiólogo de forma sistemática al interpretar un electrocardiograma: duración de los complejos, intervalos clave y corrección del QT según la frecuencia cardiaca.

Análisis de errores

Ningún sistema de clasificación es perfecto. En este prototipo se observaron tres fuentes principales de error (tanto falsos positivos como falsos negativos):

- **Señal ruidosa o con deriva:** cuando el paciente se mueve, se despegan ligeramente los electrodos o hay interferencias eléctricas, la señal se contamina. Esto puede distorsionar los intervalos PR, QRS y QT detectados automáticamente, llevando al modelo a clasificar erróneamente la condición del paciente.
- **Valores en el límite fisiológico:** existen pacientes cuyos parámetros se encuentran en la frontera entre lo normal y lo anormal (por ejemplo, QTc ligeramente prolongado). En estos casos, incluso entre especialistas puede existir incertidumbre, lo que también se refleja en el comportamiento del modelo.
- **Ritmos altamente irregulares:** en situaciones como la fibrilación auricular, la identificación de ondas P y la regularidad del ritmo se ven comprometidas. Aunque el modelo utiliza información del RR y del QRS, algunos de estos registros pueden quedar mal clasificados, especialmente si hay pocos ejemplos en el conjunto de entrenamiento.

A partir de estos errores se proponen varias estrategias de mitigación:

- **Refuerzo del control de calidad de la señal:** aplicar pruebas automáticas de calidad antes de clasificar un registro (detección de saturación, deriva excesiva o pérdida de contacto) y, si es necesario, repetir la captura.

- **Normalización por sujeto:** el uso de técnicas como la normalización tipo *z-score* por sesión puede reducir diferencias de escala entre pacientes.
- **Modelos de ensamble y más datos anormales:** el uso de *ensembles* como Random Forest ha mostrado mejoras frente a modelos simples. A medida que se incorporen más registros anormales validados clínicamente, se espera una reducción adicional de la tasa de error.

Fortalezas y limitaciones

Fortalezas. El sistema desarrollado presenta varias fortalezas que lo hacen atractivo para entornos reales, especialmente en instituciones con recursos limitados:

- **Baja latencia de inferencia local:** el modelo se ejecuta en el mismo equipo donde se capturan las señales, con tiempos de respuesta del orden de milisegundos (< 5 ms), permitiendo una retroalimentación casi inmediata.
- **Transparencia clínica:** aunque el mejor desempeño se obtiene con un ensamble de árboles, la lógica se basa en reglas comprensibles sobre intervalos (QTc, QRS, RR, PR), lo que facilita la interpretación médica.
- **Portabilidad y bajo costo:** el uso de hardware abierto (OpenBCI Ganglion) y de un *stack* de software ligero (Python + PHP + MySQL) permite desplegar el sistema sin requerir infraestructura sofisticada.
- **Arquitectura *edge-server*:** la clasificación principal se ejecuta en el borde, mientras que el servidor web se encarga del almacenamiento y la visualización, haciendo al sistema robusto ante cortes temporales de conectividad.

Limitaciones. Al mismo tiempo, es importante reconocer las limitaciones actuales del prototipo:

- **Tamaño muestral limitado:** aunque se cuenta con registros etiquetados, el número efectivo utilizado en los experimentos es reducido para un sistema de aprendizaje automático.
- **Una sola derivación:** el uso de una única derivación limita la capacidad para detectar alteraciones visibles sólo en derivaciones específicas.
- **Sensibilidad reducida en ritmos extremos:** en bradicardias o taquicardias muy marcadas, la detección de ondas y el cálculo de intervalos pueden verse afectados.
- **Sesgo de centro único:** todos los datos proceden de un mismo hospital, lo que puede limitar la generalización del sistema a otros contextos clínicos.

Por estas razones, se recomienda realizar estudios de validación externa en otras sedes y, en el futuro, extender el sistema a configuraciones multicanal.

Perspectivas y trabajos futuros

A partir de lo observado, se plantean varias líneas de trabajo que pueden fortalecer el sistema:

- **Ensamblés y *boosting*.** Aunque el Random Forest ya es un ensamble, se pueden explorar técnicas de *boosting* (como Gradient Boosting o XGBoost) para mejorar la sensibilidad en pacientes anormales atípicos. Estos métodos están diseñados para poner más atención en los ejemplos que el modelo actual clasifica mal.
- **Corrección de artefactos.** La combinación de filtrado adaptativo y modelos ligeros de aprendizaje profundo puede ayudar a detectar y corregir artefactos de movimiento o ruido severo antes de que la señal se envíe al clasificador.
- **Multicanalidad.** La placa Ganglion dispone de cuatro canales. Aprovechar esta capacidad (por ejemplo, simulando derivaciones precordiales o combinando derivaciones estándar) permitiría capturar más información sobre la actividad eléctrica del corazón y mejorar la capacidad diagnóstica.
- **Expansión de la cohorte y estudios longitudinales.** A medida que se integren más pacientes —especialmente de grupos de riesgo como diabéticos, hipertensos o personas con antecedentes de eventos cardiovasculares— será posible realizar estudios longitudinales y evaluar cómo evoluciona el riesgo a lo largo del tiempo.
- **Reportes estructurados e integración con sistemas clínicos.** La generación de reportes PDF con anotaciones, así como la integración con estándares como HL7/FHIR, facilitaría la incorporación de este sistema en la historia clínica electrónica de los pacientes.

Comparación con otros sistemas y con la tesis de Sánchez Ovalle

Al comparar este prototipo con soluciones comerciales existentes, se observa un contraste importante en términos de costo, dependencia de la nube y flexibilidad. Muchos sistemas comerciales integrados de monitoreo y análisis de ECG:

- requieren una inversión inicial elevada (del orden de USD 2,000–10,000 por dispositivo),
- utilizan licenciamiento anual por encima de USD 1,000, y
- dependen de plataformas en la nube, con latencias típicas de 50 ms a 200 ms y con la necesidad de enviar la señal cruda o parcialmente procesada a un servidor externo.

Frente a este panorama, el presente trabajo muestra que, con hardware abierto (OpenBCI Ganglion) y software ligero, es posible alcanzar una precisión diagnóstica competitiva con latencias de inferencia menores a 5 ms mediante procesamiento local. Además, la señal cruda puede permanecer en el entorno del hospital, lo que ayuda a preservar la privacidad del paciente y reduce la exposición de datos sensibles en redes externas.

Comparación con Apple Watch (Series 11) y ECG clínico de 12 derivaciones.

Además de los sistemas comerciales, resulta útil contrastar el prototipo con (i) un dispositivo de consumo ampliamente difundido como Apple Watch (p. ej., Series 11) y (ii) el electrocardiograma clínico estándar de 12 derivaciones.

Apple Watch puede generar un ECG *similar a una derivación única (Lead I)* y su interpretación está orientada principalmente a ritmo y a la clasificación de fibrilación auricular; en cambio, en consultorio, el ECG estándar *usualmente es de 12 derivaciones*, registrando la actividad eléctrica desde distintos ángulos del corazón, lo cual incrementa la capacidad diagnóstica global.¹

Con el fin de explicitar criterios de comparación (cualitativos y cuantitativos) a manera de rúbrica, el Cuadro 6.3 resume los principales atributos.

Comparación con la tesis de Sánchez Ovalle (2020)

El trabajo de Sánchez Ovalle emplea un sensor AD8232 conectado a un teléfono Android para registrar la señal ECG y almacenarla en una aplicación móvil. Ese prototipo se centra en la adquisición de la señal para su análisis posterior, y según lo reportado, no cuenta con una validación clínica formal por parte de cardiólogos ni con un módulo completo de clasificación automática integrado al flujo de trabajo hospitalario.

En contraste, el sistema presentado en esta tesis:

- realiza un **filtrado y detección de ondas en tiempo real** (*streaming*) mediante Python y BrainFlow,
- integra una **revisión ciega con cardiólogos** del Hospital Central de San Luis Potosí, que han evaluado la claridad de la traza y la utilidad clínica del prototipo,
- incorpora **mecanismos de integridad y seguridad** (cálculo de resúmenes SHA-256 y uso de TLS 1.3 en el servidor web) para proteger los datos almacenados, y
- ofrece una **visualización web interactiva** basada en Bootstrap y Plotly, con posibilidad de exportar trazas y métricas para su inclusión en reportes clínicos.

En síntesis, mientras que el trabajo de Sánchez Ovalle representa un primer acercamiento a la adquisición móvil de señales ECG, el sistema desarrollado en esta tesis amplía ese enfoque hacia un *pipeline* completo: captura, procesamiento en tiempo real, clasificación automática basada en modelos validados, seguridad de la información y visualización amigable para el personal médico.

Buenas prácticas de reporte y reproducibilidad

Un aspecto clave en proyectos que combinan *machine learning* y datos clínicos es la capacidad de reproducir los resultados. Es decir, que otros investigadores o el propio equipo,

¹Apple Support, “Take an ECG with the ECG app on Apple Watch” (publicado 15/sep/2025): <https://support.apple.com/120278>. En particular: la app genera un ECG similar a derivación I y el ECG estándar de consultorio suele ser de 12 derivaciones; la app no detecta infarto y permite exportar un PDF.

Tabla 6.3: Rúbrica de comparación (criterios cualitativos y cuantitativos) entre el prototipo propuesto, Apple Watch (ECG) y el ECG clínico de 12 derivaciones.

Tipo: CN = criterio numérico (cuantitativo); CL = criterio lógico (cualitativo).

Criterio	Tipo	Prototipo (Ganglion + Python + PHP)	Apple Watch (Series 11 / ECG)	ECG clínico (12 derivaciones)
Número de derivaciones	CN	1 derivación equivalente (p. ej., II)	1 derivación (similar a Lead I)	12 derivaciones
Objetivo principal	CL	Apoyo al cardiólogo: intervalos + sugerencia Normal/Anormal	Ritmo y detección de AFib en registros puntuales	Diagnóstico clínico integral (múltiples ejes/ángulos)
Portabilidad	CL	Alta (equipo ligero + PC local)	Muy alta (wearable)	Media (equipo clínico; típicamente en clínica/hospital)
Dependencia de nube	CL	No necesaria (procesamiento local)	No necesariamente (procesamiento/control en el ecosistema del dispositivo)	No necesaria
Privacidad (señal cruda)	CL	La señal puede permanecer en el entorno hospitalario	Exportación típica como PDF; acceso/control vía app	Señal clínica bajo control del hospital
Exportación / compartición	CL	Reportes/figuras y métricas; trazabilidad en BD	Exporta PDF para el médico	Reporte clínico estándar; integración HIS/EMR
Latencia de inferencia	CN	< 5 ms (inferencia local del clasificador)	N/D (no se reporta como métrica clínica del dispositivo)	N/A (interpretación clínica; sin inferencia ML por defecto)
Capacidad para descartar infarto	CL	No es objetivo del clasificador (Normal/Anormal)	No detecta infarto (limitación explícita)	Sí aporta evidencia diagnóstica relevante
Flexibilidad / extensibilidad	CL	Alta (pipeline modificable; modelos interpretables)	Limitada (plataforma cerrada)	Media (depende del equipo/software clínico)
Costo relativo	CL	Bajo-medio (hardware abierto + stack ligero)	Consumo (dependiente del modelo)	Medio-alto (equipamiento y operación clínica)

en el futuro, puedan volver a ejecutar el proceso y obtener métricas similares. Para ello, en esta tesis se adoptaron las siguientes buenas prácticas:

- **Semillas fijas y registro de versión.** En todos los procesos de muestreo y validación cruzada se fijan semillas aleatorias, de modo que los pliegues (*folds*) sean reproducibles. Además, se registra la versión del código y del modelo utilizado.
- **Bundle completo del modelo.** Cada modelo entrenado se guarda en un paquete que incluye: el objeto de `scikit-learn`, el umbral `threshold` utilizado para decisión binaria, la lista de columnas `feature_cols`, los parámetros `params` del entrenamiento, el esquema de validación `cv_folds` y la política de muestreo por clase (`per_class`). Este paquete permite cargar el modelo y repetir la evaluación con nuevas muestras.
- **Curvas y tablas guardadas.** Las figuras generadas (curvas ROC/PR, matriz de confusión, importancia de variables y, en el futuro, estudios de *ablation*) se guardan en archivos de imagen, y sus métricas asociadas se almacenan en la base de datos mediante un campo JSON. De este modo, la información no se pierde y puede consultarse más adelante.
- **Scripts de extracción y evaluación.** Se dispone de scripts y archivos de automatización (por ejemplo, `Makefile` o archivos `.bat` en Windows) que encapsulan el proceso completo: descargar los datos del servidor, entrenar el modelo, generar métricas y subir los resultados a la base de datos. Esto facilita que cualquier miembro del equipo pueda repetir el entrenamiento siguiendo un procedimiento estándar.

Estas prácticas no solo aumentan la calidad científica del proyecto, sino que también ayudan a que el sistema pueda mantenerse y evolucionar en el tiempo.

Listados útiles para la evaluación (plantillas)

En esta sección se incluyen fragmentos de código en Python que sirven como plantillas para extender el análisis. Aunque no todos se emplearon en los experimentos reportados, se dejan documentados para facilitar futuras evaluaciones más completas.

Cálculo de IC 95 % por *bootstrap* (plantilla)

El siguiente listado ilustra cómo se podría calcular un intervalo de confianza (IC) del 95 % para métricas como la exactitud (ACC) o el `Recall(1)` de la clase Anormal, utilizando *bootstrap* estratificado.

Listing 6.1: Bootstrap estratificado para IC 95 % de ACC y `Recall(1)`

```
import numpy as np

def bootstrap_ci(y_true, y_pred, metric_fn, B=1000, rng=42):
    rng = np.random.default_rng(rng)
    y_true = np.asarray(y_true)
    y_pred = np.asarray(y_pred)
    idx = np.arange(len(y_true))
```

```

vals = []

for _ in range(B):
    # muestreo estratificado: re-sample dentro de cada clase
    idx0 = idx[y_true == 0]
    idx1 = idx[y_true == 1]
    s0 = rng.choice(idx0, size=len(idx0), replace=True)
    s1 = rng.choice(idx1, size=len(idx1), replace=True)
    i = np.concatenate([s0, s1])
    vals.append(metric_fn(y_true[i], y_pred[i]))

vals = np.sort(vals)
mean = float(np.mean(vals))
low = float(vals[int(0.025 * B)])
high = float(vals[int(0.975 * B)])
return mean, low, high

```

Curvas ROC/PR con OOF (plantilla)

A continuación se muestra cómo obtener las curvas ROC y Precisión–Recall a partir de las probabilidades OOF (*out-of-fold*) generadas por el modelo.

Listing 6.2: Curvas ROC y PR con OOF

```

from sklearn.metrics import roc_curve, precision_recall_curve, auc

# y_true: etiquetas verdaderas OOF (0=Normal, 1=Anormal)
# p_hat: probabilidades OOF para la clase 1 (Anormal)
fpr, tpr, thr = roc_curve(y_true, p_hat)
prec, rec, _ = precision_recall_curve(y_true, p_hat)

auc_roc = auc(fpr, tpr)
auc_pr = auc(rec, prec)

```

Selección de umbral óptimo (plantilla)

Finalmente, se presenta un ejemplo de cómo calcular un umbral óptimo para convertir probabilidades en decisiones binarias, ya sea mediante el índice de Youden o buscando el umbral que maximiza la exactitud.

Listing 6.3: Cálculo de t^* por Youden y ACC-max

```

import numpy as np

def youden_threshold(fpr, tpr, thr):
    j = tpr - fpr
    return float(thr[np.argmax(j)])

def accmax_threshold(y_true, p_hat):
    ts = np.linspace(0, 1, 501)
    accs = []

    for t in ts:

```

```
y_pred = (p_hat >= t).astype(int)
acc = (y_pred == y_true).mean()
accs.append((t, acc))

t_best, _ = max(accs, key=lambda x: x[1])
return float(t_best)
```

Estos listados constituyen una base para que, en etapas posteriores del proyecto, se puedan incorporar intervalos de confianza, análisis de sensibilidad por subgrupos y ajustes más finos de los umbrales de decisión.

Resumen del capítulo

En este capítulo se discutieron los resultados del clasificador Normal/Anormal desarrollado a partir de señales ECG capturadas con la placa OpenBCI Ganglion. Se mostró que el modelo Random Forest alcanza una exactitud OOF de aproximadamente 80.95% y un AUC-ROC cercano a 0.872 en un conjunto balanceado de 90 registros (45 normales y 45 anormales), lo que indica una capacidad razonable para distinguir entre pacientes con trazas normales y anormales.

Se destacó que las variables más importantes para el modelo (QTc, QRS, RR y PR) coinciden con criterios bien establecidos en la práctica cardiológica, lo que apoya la interpretabilidad clínica del sistema. También se analizaron los errores más frecuentes (ruido, límites fisiológicos y ritmos altamente irregulares) y se propusieron estrategias concretas para mitigarlos.

Asimismo, se presentaron las fortalezas del prototipo (baja latencia, transparencia, bajo costo y arquitectura *edge-server*) y sus limitaciones actuales (tamaño muestral, una sola derivación, sensibilidad reducida en casos extremos y sesgo de centro único). Finalmente, se esbozaron las principales líneas de trabajo futuro y se comparó la solución propuesta con sistemas comerciales y con la tesis de Sánchez Ovalle, resaltando la evolución desde un enfoque de adquisición móvil hacia un sistema integral de captura, clasificación, seguridad y visualización orientado al apoyo del diagnóstico de enfermedades cardiovasculares.

Capítulo 7

Conclusiones y trabajo futuro

Este capítulo resume las principales aportaciones del trabajo, analiza el grado de cumplimiento de los objetivos y propone una ruta clara para continuar el desarrollo del sistema. La intención es que cualquier persona interesada —personal técnico, médico o de salud— pueda entender qué se logró, qué falta por hacer y cómo podría evolucionar el proyecto.

7.1. Síntesis de aportaciones

A lo largo de esta tesis se demostró que es viable construir un sistema de apoyo al diagnóstico cardiovascular, de bajo costo y técnicamente asequible, orientado a atención primaria y a contextos donde la conectividad a Internet puede ser limitada o inestable. En consonancia con las aportaciones planteadas en el Capítulo 1, las aportaciones principales son:

- **Adquisición accesible y base de datos propia, controlada y reproducible.** Se integró la placa **OpenBCI Ganglion** con **Python** (vía **BrainFlow**) para capturar ECG de una sola derivación a 200 Hz y se definió un protocolo de posicionamiento y captura documentado y repetible, con el que se construyó una base de datos experimental.
- **Preprocesamiento digital y extracción automática de características clínicas.** Se diseñó un flujo de limpieza de señal con filtrado en *cero fase* (detrending, pasa-banda 0.5–45 Hz y notch a 60 Hz) y se automatizó el cálculo de **RR**, **PR**, **QRS**, **QT**, **ST**, **QTc** (Bazett) y **BPM**. Cada traza de 30 s se resume en un vector de características clínicamente interpretable por el especialista.
- **Clasificación interpretable y evaluación robusta con umbral explícito.** Se entrenaron modelos **CART** y **Random Forest** usando validación cruzada estratificada (predicciones *out-of-fold*). Se documentó la selección del umbral (*threshold*) y su interpretación clínica (p. ej., equilibrio sensibilidad/especificidad), dejando el sistema auditable y ajustable según la prioridad clínica.
- **Plataforma *edge-server* para operación clínica, trazabilidad y despliegue.** Se implementó una arquitectura híbrida tolerante a fallos donde adquisición, filtrado e infe-

rencia se ejecutan localmente, mientras que el servidor gestiona el repositorio y la consulta de evidencias. En particular, se aportó:

- **Visualización clínica y trazabilidad:** interfaz web para consultar señales, métricas e indicadores; cada captura asociada a un *folio* único para auditoría.
- **Seguridad e integridad:** transporte cifrado (**TLS 1.3**), verificación de integridad (**SHA-256**) y control de acceso por roles; pseudonimización para reducir exposición de datos identificables.
- **Optimización del almacenamiento experimental:** sustitución de campos *BLOB* por rutas a archivos y un campo JSON (*metrics_json*) para evidencias y métricas, facilitando respaldos y reduciendo tamaño de base de datos.
- **Reproducibilidad:** registro de versiones, semillas fijas, empaquetado del modelo (columnas, parámetros y umbral) y *scripts* para repetir evaluación y generar tablas/curvas sin pasos manuales.

Hallazgos transversales (para orientar el trabajo futuro).

1. La mayor ganancia de desempeño provino de **mejorar la calidad/consistencia de la señal y la trazabilidad** (preprocesamiento bien parametrizado + evidencia por folio), más que de incrementar complejidad del modelo.
2. El factor más valioso para aumentar sensibilidad y utilidad clínica es **incrementar casos anormales bien validados** (etiquetado clínico consistente), ya que los cambios puramente algorítmicos tienen impacto limitado con muestras pequeñas.

7.2. Cumplimiento de objetivos

El objetivo general planteado al inicio de esta tesis fue:

Objetivo general: Desarrollar un sistema de apoyo al diagnóstico cardiovascular que integre la adquisición de la señal ECG, su procesamiento y modelos interpretables, junto con una visualización web orientada al personal de salud.

De forma resumida, el cumplimiento se describe así:

1. **Captura y limpieza.** Se capturaron señales a 200 Hz con un *pipeline* reproducible de filtrado que mejora relación señal-ruido y estabiliza la línea de base. Se documentaron parámetros y recomendaciones operativas (p. ej., reducir bucles de tierra operando en batería cuando sea necesario).
2. **Detección de ondas y cálculo de características.** Se implementó un esquema de detección de ondas (P, QRS, T) y cálculo de intervalos/variables derivadas relevantes para la práctica clínica, incluyendo QTc y BPM.
3. **Clasificación y política de umbral.** Se construyó un clasificador binario *Normal/A-normal* (CART/RF) y se definió una política de umbral explícita para ajustar el *trade-off* sensibilidad/especificidad según el contexto (tamizaje vs. confirmación).

4. **Despliegue, trazabilidad y visualización web.** Se integró un flujo completo Python–servidor (PHP/MySQL) con almacenamiento estructurado, métricas y visualización web para revisión clínica y seguimiento por folio.

En conjunto, el objetivo general se considera **cumplido**: se cuenta con un sistema funcional *end-to-end* desde la captura hasta la visualización y el apoyo al diagnóstico.

7.3. Hallazgos principales

Durante el desarrollo y evaluación del sistema, se identificaron hallazgos relevantes:

- **El preprocesamiento determinó la estabilidad del sistema.** El filtrado en *cero fase* y las ventanas centradas en picos R fueron claves para estabilizar detecciones y evitar sesgos por *phase lag*. Cuando el filtrado se parametriza mal, se degradan las características (intervalos) y el desempeño del clasificador cae de forma directa.
- **Las variables que “pesan” coinciden con criterios clínicos.** QTc, ancho de QRS, y marcadores derivados de RR/PR aportaron más a la separación Normal/Anormal, alineándose con conceptos clínicos (repolarización, conducción y regularidad del ritmo). Esto fortalece la interpretabilidad del sistema.
- **La política de umbral sí impactó el desempeño “útil” del prototipo.** Más allá del AUC, ajustar el umbral permitió controlar el tipo de error: por ejemplo, priorizar sensibilidad para reducir falsos negativos en escenarios de tamizaje, o equilibrar sensibilidad/exactitud para reducir alarmas innecesarias. En la práctica, este ajuste fue una de las palancas más claras para mejorar el comportamiento del árbol/bosque en un conjunto pequeño.
- **Las etiquetas clínicas son esenciales para mejorar.** La disponibilidad de `diagnostico_doctor` y la etiqueta binaria `condicion` permitió entrenar, auditar y corregir el sistema; sin estas etiquetas, el flujo quedaría limitado a un análisis técnico sin conexión con el criterio clínico.

7.4. Optimización

Aunque el sistema ya es usable con la configuración actual, se han aplicado y propuesto varias estrategias de optimización para mejorar el rendimiento y preparar el terreno para escenarios con mayor volumen de datos:

- **Vectorización con NumPy y filtrado en cero fase.** Los cálculos sobre la señal (filtros, derivadas, intervalos) se realizan sobre vectores y arreglos completos, evitando bucles innecesarios en Python. El filtrado en cero fase se aplica solo una vez por segmento de señal, aprovechando las rutinas ya optimizadas de BrainFlow o `scipy`.

- **Uso futuro de Cython.** En etapas posteriores se considera migrar las partes más intensivas (detección de picos R e intervalos por latido) a módulos compilados con **Cython**. Esta estrategia puede reducir los tiempos de procesamiento entre 3 y 5 veces en equipos modestos, manteniendo la misma lógica de alto nivel.
- **Batching de predicciones y reutilización de conexiones HTTP.** Para la etapa de predicción, el módulo en Python agrupa varios registros pendientes y los procesa en lote, en lugar de hacer una llamada a la API por cada estudio. Además, se reutilizan sesiones HTTP con `requests.Session`, lo que disminuye la sobrecarga de abrir y cerrar conexiones al servidor.
- **Optimización en base de datos.** En la base de datos se han definido índices por `folio`, `fecha` y `condicion`, lo que acelera las consultas más frecuentes (búsqueda de estudios por fecha o filtrado por Normal/Anormal). De forma complementaria, se pueden comprimir o archivar los CSV más antiguos y establecer políticas de retención y depuración para mantener el tamaño de la base de datos bajo control.

Estas decisiones permiten que, si en el futuro se incrementa el número de estudios o se añaden más derivaciones, el sistema pueda escalar de manera gradual sin rediseñarse desde cero.

7.5. Trabajo futuro

Las líneas de trabajo futuro se agrupan en ejes técnicos y clínicos, retomando también oportunidades identificadas en capítulos anteriores (robustez a artefactos, multicanalidad, expansión de cohorte, calibración, subgrupos, reproducibilidad e integración clínica).

Señal y adquisición

- **ECG multiderivación (3/6/12 derivaciones).** Extender el hardware/interfaz para múltiples canales sincronizados y ajustar el modelo/BD para registrar derivaciones con trazabilidad. Como línea incremental, iniciar con 3 derivaciones (triángulo de Einthoven) y evaluar el beneficio clínico y estadístico antes de escalar a 6/12.
- **Estrategias de expansión de derivaciones con mínima invasividad.** Explorar arneses o configuraciones de electrodos de colocación rápida (p. ej., plantillas/guías de posición) para reducir errores de colocación y facilitar capturas repetibles en campo.
- **Mayor robustez al movimiento.** Incorporar control de artefactos (p. ej., filtros adaptativos, separación de fuentes) y explorar modelos ligeros (incluyendo enfoques de *boosting* o redes compactas) para detección/corrección de ruido antes de clasificar.
- **Control de calidad automático.** Definir índices por ventana (SNR, deriva, porcentaje de notch efectivo) y banderas operativas (p. ej., `requiere_repetir`) para advertir al operador cuando la captura no cumple umbrales mínimos.

- **Protocolos de captura para uso no especializado.** Diseñar un flujo guiado de captura (pasos, tiempos y verificación automática de calidad) orientado a personal no experto, para mejorar repetibilidad y reducir registros inválidos.

Características y personalización clínica

- **Normalización por sujeto o sesión.** Evaluar *z-score* por registro/sujeto y escalados robustos, midiendo impacto en AUC y Recall(1) para la clase Anormal.
- **Características avanzadas.** Integrar HRV (RMSSD, pNN20/pNN50), rasgos morfológicos del QRS, y métricas adicionales de repolarización (QT dinámico), manteniendo interpretabilidad.
- **Ventanas adaptadas a BPM/edad.** Ajustar ventanas de búsqueda de ondas en función de la frecuencia cardiaca y (si aplica) edad, para mejorar desempeño en ritmos muy rápidos/lentos o irregulares.
- **Sugerencias clínicas más específicas (más allá de Normal/Anormal).** Manteniendo un enfoque interpretables, explorar una segunda etapa de clasificación por *familias* (p. ej., taquicardia, bradicardia, irregularidad marcada) como apoyo inicial al especialista, cuidando el riesgo de sobrefirmación.

Modelado, calibración y evaluación

- **Comparativos robustos de modelos.** Comparar CART/RF contra *gradient boosting* (Gradient Boosting, XGBoost, etc.) evaluando AUC, PR-AUC en desbalance, calibración (Brier, curvas de confiabilidad) e intervalos de confianza vía *bootstrap*.
- **Estudios de *ablation*.** Ejecutar *ablation* sistemático (remover QTc, QRS, RR, PR) para cuantificar pérdida relativa y respaldar decisiones de ingeniería/clinicia.
- **Subgrupos y equidad.** Evaluar desempeño por sexo, rangos de edad y condiciones de señal; detectar si el sistema funciona peor en segmentos específicos y proponer correcciones.
- **Validación multisede y longitudinal.** Extender el estudio a otras sedes y, cuando sea posible, realizar seguimiento longitudinal para evaluar estabilidad del riesgo y generalización fuera del centro de origen.
- **Estimación de incertidumbre y umbrales dependientes del contexto.** Incorporar señales de incertidumbre (p. ej., probabilidad calibrada/intervalos) y explorar umbrales distintos para escenarios (*tamizaje* vs. clínica), priorizando sensibilidad cuando corresponda.

Comparación con dispositivos comerciales y ECG clínico

- **Rúbrica formal de comparación.** Implementar y reportar una rúbrica con criterios **cualitativos** (alcance clínico, portabilidad, privacidad, integración a flujo clínico) y **cuan-**

titativos (latencia, costos, métricas de desempeño) para comparar el prototipo frente a: (i) Apple Watch (ECG) y (ii) ECG clínico de 12 derivaciones, explicitando limitaciones de cada enfoque.

- **Estudios de concordancia.** Cuando sea viable, comparar intervalos (PR, QRS, QT/QTc) y detecciones de eventos (picos R, irregularidad) contra referencia clínica, reportando concordancia y sesgos.

Plataforma, movilidad y despliegue en campo

- **Cliente móvil (Android/iOS) como interfaz principal o alternativa.** Implementar una aplicación móvil que reemplace la PC local para: (i) iniciar/detener captura, (ii) visualizar señal y calidad, (iii) ejecutar el procesamiento e inferencia en el dispositivo (*edge*), y (iv) registrar metadatos clínicos mínimos.
- **Modo *offline* y sincronización diferida.** Diseñar almacenamiento local cifrado en el dispositivo (registros + resultados) y una cola de subida que sincronice con el servidor cuando exista conectividad, evitando pérdida de datos en entornos con internet intermitente.
- **Uso en zonas rurales y brigadas de salud.** Adaptar el flujo para contextos de baja infraestructura: paquetes ligeros, captura guiada, validación automática de calidad, y envío de reportes cuando haya señal, con el objetivo de apoyar *tamizaje* y referencia oportuna.
- **Optimización del código crítico.** Migrar partes pesadas a Cython/Numba o técnicas similares para reducir tiempo por registro y mejorar respuesta en equipos modestos.
- **Comunicación robusta con el servidor.** Implementar colas de sincronización (subida diferida), sesiones HTTP persistentes y reintentos controlados para evitar bloqueos durante captura en tiempo real.
- **Reportes clínicos en PDF e integración.** Generar reportes PDF con trazas anotadas, intervalos, umbral aplicado y resumen de métricas; explorar integración con estándares (HL7/FHIR) para facilitar adopción.
- **Gestión operativa y trazabilidad.** Incluir bitácoras de auditoría, versionado de modelos por paciente (*modelo-umbral-features*) y control de roles para soporte en entornos clínicos.

Ciencia abierta y reproducibilidad

- **Versionado de artefactos.** Mantener historial de modelos, umbrales y columnas (*bundle*) y, donde sea éticamente posible, publicar scripts y artefactos sin datos sensibles.
- **Documentación del dataset.** Elaborar *datasheets* del conjunto de datos (campos, procedencia, criterios de inclusión/exclusión, limpieza), facilitando auditoría y replicación.

- **Automatización de evaluación.** Consolidar scripts (p. ej., `Makefile` / `.bat`) para repetir entrenamiento/validación, curvas ROC/PR y tablas, registrando semillas y dependencias (`requirements.txt`).

7.6. Recomendaciones operativas

Además de las líneas de investigación, se proponen recomendaciones para uso cotidiano:

- **Monitoreo continuo de calidad.** Mostrar indicadores simples (porcentaje de capturas “OK”, SNR/deriva promedio) y alertar cuando caigan bajo umbrales definidos.
- **Reentrenos por hitos clínicos.** Reentrenar al acumular nuevos casos anormales validados y conservar versiones de modelos con sus métricas y umbral asociado.
- **Auditoría clínica sistemática.** Revisar falsos positivos/falsos negativos para ajustar política de umbral y reglas operativas (p. ej., `requiere_revision`).
- **Gestión responsable de datos.** Definir plan de retención, anonimización y respaldos cifrados conforme a normatividad institucional aplicable.

7.7. Cierre

En esta tesis se demostró que es posible construir un sistema **end-to-end**, asequible y **explicable** para apoyar el diagnóstico de enfermedades cardiovasculares a partir de un ECG de una sola derivación. La combinación de (i) adquisición bien parametrizada, (ii) procesamiento reproducible y (iii) un clasificador interpretable con desempeño aceptable permite obtener información útil para apoyar la revisión clínica, especialmente en contextos con recursos limitados.

Las líneas de trabajo futuro —mejora de calidad automática, expansión y validación multisede, comparación formal con Apple Watch y ECG de 12 derivaciones, calibración, subgrupos e integración a reportes clínicos— ofrecen una ruta concreta para escalar el sistema de forma transparente y auditable.

En última instancia, detrás de cada registro ECG hay una persona con síntomas y preocupaciones. El objetivo de este trabajo ha sido avanzar hacia herramientas tecnológicas que, sin sustituir al médico, ayuden a una detección más oportuna de problemas cardiovasculares con un costo razonable y apoyándose en soluciones abiertas y reproducibles.

Bibliografía

- [1] W. H. Organization, “Cardiovascular diseases (cvds),” 2023. Consultado el 5 julio 2025.
- [2] G. A. Roth *et al.*, “Global burden of cardiovascular diseases and risk factors, 1990–2019,” *Journal of the American College of Cardiology*, vol. 76, no. 25, pp. 2982–3021, 2020.
- [3] K. J. Foreman, N. Marquez, A. Dolgert, K. Fukutaki, N. Fullman, M. McGaughey, M. A. Pletcher, A. E. Smith, K. Tang, C.-W. Yuan, *et al.*, “Forecasting life expectancy, years of life lost, and all-cause and cause-specific mortality for 250 causes of death: reference and alternative scenarios for 2016–2040 for 195 countries and territories,” *The Lancet*, vol. 392, no. 10159, pp. 2052–2090, 2018. Consultado el 5 julio 2025.
- [4] I. N. de Estadística y Geografía (INEGI), “Estadísticas de defunciones registradas EDR 2024, primer trimestre.” Boletín de prensa EDR/2024_1erT, 2024. Acceso: 5 julio 2025.
- [5] S. de Salud de México, “Informe económico y financiero del sector salud 2023.” Documento técnico, 2023. Acceso: 5 julio 2025.
- [6] G. Stevens, R. H. Dias, K. Thomas, J. A. Rivera, R. Martorell, S. Barquera, and J. L. Bobadilla, “The economic burden of hypertension, heart failure, myocardial infarction, and atrial fibrillation in Mexico,” *Value in Health Regional Issues*, vol. 17, pp. 33–39, 2018.
- [7] I. N. de Salud Pública (INSP), “Encuesta nacional de salud y nutrición 2022: Resultados sobre factores de riesgo cardiometabólicos en México.” Reporte técnico, 2023. Consultado el 5 julio 2025.
- [8] U. N. D. of Economic and P. D. Social Affairs, “World population prospects 2022: Summary of results.” UN DESA/POP/2022/TR/NO. 3, 2022. Consultado el 5 julio 2025.
- [9] E. Wilkins, N. Townsend, and other contributors, “European cardiovascular disease statistics 2017.” European Heart Network and European Society of Cardiology report, 2017. Consultado el 5 julio 2025.
- [10] P. Ssentongo, A. E. Ssentongo, E. S. Heilbrunn, V. M. Chinchilli, and C. I. Paules, “Prevalence of cardiovascular comorbidities in coronavirus disease 2019 (COVID-19): a meta-analysis,” *PLOS ONE*, vol. 15, no. 8, p. e0238215, 2020.

- [11] W. H. Organization, “Saving lives, spending less: the case for investing in noncommunicable disease prevention.” WHO report, 2021. Accessed 5 Jul 2025.
- [12] S. Sharma, J. A. Drezner, A. Baggish, M. Papadakis, M. G. Wilson, J. M. Prutkin, A. La Gerche, M. J. Ackerman, M. Borjesson, J. C. Salerno, *et al.*, “International recommendations for electrocardiographic interpretation in athletes,” *European Heart Journal*, vol. 38, no. 31, pp. 2973–2989, 2017.
- [13] W. Einthoven, “Le télécardiogramme,” *Archives Internationales de Physiologie*, vol. 4, pp. 132–164, 1906.
- [14] R. J. Prineas, R. S. Crow, and H. Blackburn, *The Minnesota Code Manual of Electrocardiographic Findings*. Littleton, MA: J. Wright PSG, 1982.
- [15] L. Tarassenko and D. Clifton, “Remote monitoring of vital signs—current status and future challenges,” *BioMedical Engineering OnLine*, vol. 17, no. 1, p. 124, 2018.
- [16] M. V. Perez, K. W. Mahaffey, H. Hedlin, J. S. Rumsfeld, A. Garcia, T. Ferris, V. Balasubramanian, A. M. Russo, A. Rajmane, J. Cheung, *et al.*, “Large-scale assessment of a smartwatch to identify atrial fibrillation,” *New England Journal of Medicine*, vol. 381, no. 20, pp. 1909–1917, 2019.
- [17] E. Hatef, R. F. Wilson, S. M. Hannum, A. Zhang, H. Kharrazi, J. P. Weiner, S. A. Davis, and K. A. Robinson, “Use of telehealth during the covid-19 era: Systematic review.” AHRQ Evidence Report 23-EHC005, 2023. Consultado el 5 julio 2025.
- [18] D. Sánchez Ovalle, “Aplicación móvil para la adquisición y procesamiento de señales electrocardiográficas,” Master’s thesis, Universidad Autónoma de San Luis Potosí, 2020.
- [19] J. E. Hall, *Guyton and Hall Textbook of Medical Physiology*. Elsevier, 14 ed., 2021.
- [20] R. H. Anderson and S. Y. Ho, “The morphology of the cardiac conduction system,” *Heart*, vol. 89, no. 10, pp. 1244–1248, 2003.
- [21] B. Surawicz and T. K. Knilans, *Chou’s Electrocardiography in Clinical Practice*. Saunders, 5 ed., 2001.
- [22] K. Thygesen, J. S. Alpert, A. S. Jaffe, B. R. Chaitman, J. J. Bax, D. A. Morrow, H. D. White, and the Executive Group on behalf of the Joint European Society of Cardiology/American College of Cardiology/American Heart Association/World Heart Federation Task Force for the Universal Definition of Myocardial Infarction, “Fourth universal definition of myocardial infarction (2018),” *Journal of the American College of Cardiology*, vol. 72, no. 18, pp. 2231–2264, 2018.
- [23] P. Kirchhof, S. Benussi, D. Kotecha, A. Ahlsson, D. Atar, B. Casadei, H.-C. Diener, H. Heidbuchel, J. Hendriks, G. Hindricks, *et al.*, “2020 esc guidelines for the diagnosis and management of atrial fibrillation,” *European Heart Journal*, vol. 42, no. 5, pp. 373–498, 2020.

- [24] D. P. Zipes, J. Jalife, and W. G. Stevenson, *Cardiac Electrophysiology: From Cell to Bedside*. Elsevier, 8 ed., 2022.
- [25] R. O. Bonow and C. M. Otto, *Valvular Heart Disease: A Companion to Braunwald's Heart Disease*. Elsevier, 5 ed., 2020.
- [26] P. W. Macfarlane and T. D. V. Lawrie, *Comprehensive Electrocardiology*. Springer, 2 ed., 2011.
- [27] J. F. García-Lara and J. A. Martínez-Sánchez, “Electrocardiographic patterns and arrhythmia detection: A review of ecg signal processing,” *Journal of Medical Systems*, vol. 44, no. 9, p. 160, 2020.
- [28] M. E. Josephson, *Clinical Cardiac Electrophysiology: Techniques and Interpretations*. Wolters Kluwer, 4 ed., 2015.
- [29] R. E. Klabunde, *Cardiovascular Physiology Concepts*. Lippincott Williams & Wilkins, 2 ed., 2012.
- [30] D. P. Zipes and J. Jalife, *Electrophysiology of the Heart*. Elsevier, 5 ed., 2019.
- [31] J. S. Steinberg, N. Varma, I. Cygankiewicz, P. Aziz, P. Balsam, A. Baranchuk, D. Cantillon, P. Dilaveris, S. Dubner, N. El-Sherif, *et al.*, “2017 ishne-hrs expert consensus statement on ambulatory ecg and external cardiac monitoring/telemetry,” *Heart Rhythm*, vol. 14, no. 7, pp. e55–e96, 2017.
- [32] T. Phalen and B. Aehlert, *ECG de 12 derivaciones en los Síndromes Coronarios Agudos*. Bogotá: AMOLCA, 4 ed., 2021.
- [33] B. Developers, “Brainflow data filter module.” <https://brainflow.readthedocs.io/en/stable/filter.html>, 2023. Consultado el 5 julio 2025.
- [34] G. D. Clifford, F. Azuaje, and P. E. McSharry, *Advanced Methods and Tools for ECG Data Analysis*. Artech House, 2006.
- [35] O. LLC, “Ganglion board—technical specifications.” OpenBCI Documentation, 2023. Consultado el 5 julio 2025.
- [36] B. Developers, “Brainflow api documentation.” <https://brainflow.readthedocs.io/en/stable/api.html>, 2023. Consultado el 5 julio 2025.
- [37] B. Developers, “Brainflow examples.” <https://brainflow.readthedocs.io/en/stable/examples.html>, 2023. Consultado el 5 julio 2025.
- [38] J. Pan and W. J. Tompkins, “A real-time qrs detection algorithm,” *IEEE Transactions on Biomedical Engineering*, vol. 32, no. 3, pp. 230–236, 1985.
- [39] P. S. Hamilton, “Open source ecg analysis software,” in *Computers in Cardiology*, pp. 101–104, 2002.

- [40] P. Laguna, R. G. Mark, A. Goldberg, and G. B. Moody, “Automatic detection of qrs boundaries in multilead ecg signals: Validation with the cse database,” *Computers and Biomedical Research*, vol. 27, no. 1, pp. 45–60, 1994.
- [41] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Belmont, CA: Wadsworth International Group, 1984.
- [42] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [43] W. J. Youden, “Index for rating diagnostic tests,” *Cancer*, vol. 3, no. 1, pp. 32–35, 1950.
- [44] F. Gustafsson, “Determining the initial states in forward-backward filtering,” *IEEE Transactions on Signal Processing*, 1996.
- [45] L. Sörnmo and P. Laguna, *Bioelectrical Signal Processing in Cardiac and Neurological Applications*. Academic Press, 2005.
- [46] P. Kligfield, L. S. Gettes, J. J. Bailey, R. Childers, B. J. Deal, E. W. Hancock, G. van Herpen, J. A. Kors, P. Macfarlane, D. M. Mirvis, O. Pahlm, P. Rautaharju, and G. S. Wagner, “Recommendations for the standardization and interpretation of the electrocardiogram: Part i: The electrocardiogram and its technology,” *Circulation*, vol. 115, no. 10, pp. 1306–1324, 2007.
- [47] P. M. Rautaharju, B. Surawicz, L. S. Gettes, *et al.*, “Aha/accf/hrs recommendations for the standardization and interpretation of the electrocardiogram: Part iv: The st segment, t and u waves, and the qt interval,” *Circulation*, vol. 119, no. 10, pp. e241–e250, 2009.
- [48] B. Vandenberg, E. Vandael, T. Robyns, *et al.*, “Which qt correction formulae to use for qt monitoring?,” *Journal of the American Heart Association*, vol. 5, no. 6, p. e003264, 2016.