UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL

FRAMEWORK PARA EVALUACIÓN DE MODELOS PREDICTIVOS EN SEÑALES ELECTROCARDIOGRÁFICAS CON APLICACIÓN EN LA DETECCIÓN DE ARRITMIAS

TESIS PARA OPTAR AL TÍTULO DE INGENIERO CIVIL INDUSTRIAL

GUSTAVO EMILIO AVILÉS SEGOVIA

PROFESOR GUÍA:
JUAN DOMINGO VELÁSQUEZ SILVA

MIEMBROS DE LA COMISIÓN:
FRANCISCO MOLINA JARA
JUAN CARRILLO AZÓCAR

SANTIAGO DE CHILE
2016

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE INGENIERO CIVIL INDUSTRIAL
POR: GUSTAVO EMILIO AVILÉS SEGOVIA
FECHA: 2016
PROF. GUÍA: SR. JUAN DOMINGO VELÁSQUEZ SILVA

FRAMEWORK PARA EVALUACIÓN DE MODELOS PREDICTIVOS EN SEÑALES ELECTROCARDIOGRÁFICAS CON APLICACIÓN EN LA DETECCIÓN DE ARRITMIAS

Se ha demostrado que la supervisión remota, automática y no automática, puede influir positivamente en la disminución de la taza de hospitalizaciones y en el caso de que el paciente sea hospitalizado, disminuir la duración de esta.

Comúnmente se utiliza en finanzas, meteorología, oceanografía, entre otros el uso, de técnicas de análisis de predicción de estado usando datos históricos, de esta forma se analizan diferentes implementaciones de minería de datos sobre estados o situaciones reales para luego comparar las predicciones con la realidad gestada.

Sin embargo, a pesar de la existencia de una vasta documentación sobre como diferentes modelos y técnicas de minería de datos que son capaces de identificar problemas como arritmias, fibrilaciones auriculares, entre otros, esta tecnología no ha sido masificada, a pesar de su existencia en salas de emergencia, cuidados intermedios e intensivos.

En el contexto descrito anteriormente que se gesta el proyecto de aplicar técnicas de minería de datos sobre mediciones fisiológicas, para la detección remota de problemas en pacientes. Y con esto la necesidad del desarrollo de un sistema de pruebas, que simule la aplicación en tiempo real de minería de datos sobre signos vitales capturados en tiempo real.

Para esto se fue necesario crear un modelo de aprendizaje profundo, el que detectara el punto R del latido y clasificara éste como normal o anormal. Los resultados obtenidos luego del entrenamiento y prueba del modelo, muestran que es capaz de determinar certeramente cuando un latido es normal, sin embargo el modelo no entrega la misma seguridad al momento de clasificar un latido como anormal, ya que en ocasiones éste latido es normal. Por lo que no se pudo demostrar la hipótesis de que se pueden aplicar técnicas redes neuronales diseñadas para la detección de objetos en imágenes, sirvan para la clasificación de latidos cardíacos.

Finalmente, tras el estudio de requerimientos del framework, la creación de un prototipo y su uso, se llegó a las siguientes conclusiones, el utilizar el framework posee notables falencias en el rendimiento en la etapa de entrenamiento y evaluación del modelo, el costo de elaboración y mantención de los requerimientos no cubiertos por el prototipo no se pueden justificar, haciendo el proyecto inviable, debido a la aparición de una alternativa gratuita de código libre que cubre gran parte de los beneficios del framework.

A mi Familia, Romina y Amigos.

Agradecimientos

Quiero agradecer a mis padres, por su apoyo y crianza, muchos de mis méritos fueron peleas y discusiones con ustedes en algún momento. A mis hermanos por enseñarme a ser el hermano mayor, a tener pasiencia y perdornar que te roben los dulces de la piñata, aunque los hayas escondidos dentro de un juguete con forma de pelota, en tu caja de herramientas. A decir que no, aunque te pidan mil veces que le prestes tus esquíes. A sentir el orgullo que digan que cuando grande quieren ser hermano mayor.

A mis padrinos, que a pesar de que no son mágicos, me enseñaron cosas que tus padres no harían, como a derretir el queso en el microondas sin el pan o a parsarle el dedo a la torta pro donde no se ve, entre otros.

Quiero agradecer a Romina Fernández en adelante chanchita, la chanchita que me ayudó a corregir mis problemas del lenguaje, expresión escrita, oral, ortografía, redacción, falta de conectores, dislexia, dislalia y cuanta cosa se te ocurra para justificar que escribo pésimo. A todo lo anterior y el cariño, soporte y el millón de cosas que me haz entregado y enseñado, se suma mi falta de puntualidad y las largas y supongo, fomes horas que pasate leyendo, descifrando y traduciendo esta tesis, el ñunguñó no es un lenguage muy pular.

A mis amigos, aunque hay algunos que casi no valen como amigos, si aquellos que han estado tanto tiempo acompañándote que cuantan más como familia que amigos, les doy las gracias a ustedes Carla y Matías.

Quiero agradecer a las niñas perdidas (Paulina, Constanza y Gabriela), a Betsy y tantas otras personas que hemos hecho esta aventura universitaria un camino común, que gracias a su compañía y amistad hemos echo de la cafetería el panel de maldita moda, o no te lo pongas.

To Jared Broad, for all the support and understanding about priorities, for the company and friendship that we've been building together.

Finalmente quiero agradecer a mi profesor gía por todo el respaldo y confianza que me entregó durante este proceso.

Tabla de Contenidos

Índice de Tablas

Índice de Ilustraciones

Capítulo 1

Introducción

Esta tesis aborda el estudio de la aplicación de la técnica de aprendizaje profundo para la identificación y clasificación de latidos en el ECG. Junto con esto el diseño y construcción a nivel piloto, de una arquitectura web basada en servicios, que simula condiciones reales en la aplicación de técnicas de minería de datos sobre registros clínicos electrónicos.

Mediante la utilización de metodologías de diseño de procesos se aborda el diseño de un sistema de supervisión automática, diseño que es complementado con la implementación de un modelo de detección de arritmias.

1.1. Contexto

En los últimos años se han visto importantes avances en el procesamiento, clasificación e interpretación de señales, esto en conjunto con la oferta de infraestructura como servicio, han permito que el almacenamiento y clasificación de grandes volúmenes de datos (Big Data) pueden ser realizado bajo demanda, masificando su aplicación y soportando el desarrollo de investigaciones en Web Intelligence (WI).

Según cifras de Naciones Unidas, el 43 % de la población mundial está conectada a Internet [20], lo que ha permitido el desarrollo e investigación en Health Informatics (HI) que puedan ser utilizados para supervisar pacientes de forma remota (telemedicina). Sumado a esto, la masificación de monitores multipropósito, que pueden realizar un seguimiento del pulso cardíaco, sueño, peso y actividad física que el usuario posee, han sido incorporados en equipos de uso cotidiano, realizando un constante registro médico, esto ha abierto las posibilidades a la medicina preventiva para supervisar de forma constante y alertar anormalidades en los pacientes supervisados.

La aplicación de técnicas en procesamiento y clasificación de series de tiempo fisiológicas, ha permitido desarrollar un ecosistema de aparatos inteligentes, que permiten segmentar y clasificar arritmias en pacientes internados en unidades de cuidados intermedios e intensivos [29]. Al realizar un meta análisis de los diferentes sistemas de monitoreo remoto sobre

pacientes con fallas cardíacas, se ha llegado a la conclusión que puede disminuir la taza de hospitalizaciones, y en el caso de los pacientes hospitalizados reducir la duración de las hospitalizaciones [24].

1.1.1. Situación Actual

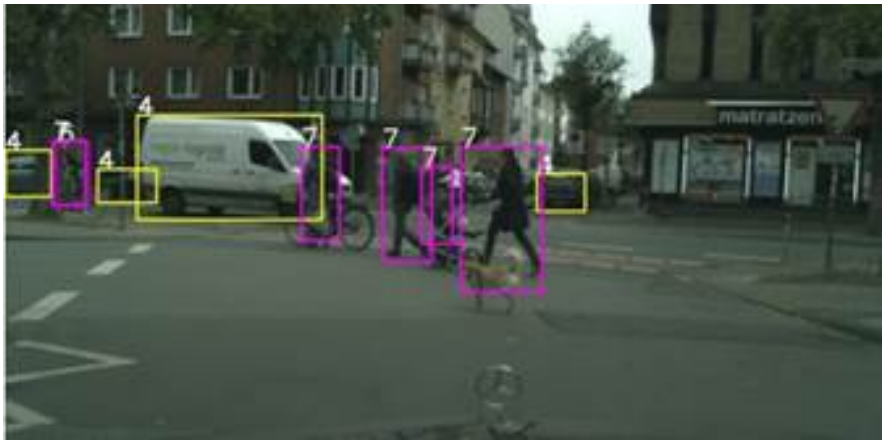

Figura 1.1: Ejemplo de clasificación y detección de objetos en una imagen, Fuente: [32]

En Chile las enfermedades cardiovasculares constituyen la primera causa de muerte, según Ministerio de Salud Chile (MINSAL) con un 30 % [8], creando oportunidades para el desarrollo de tecnologías en medicina preventiva, principalmente en grupos de riesgo para ayudar a disminuir dicho indice, salvando vidas y sus consecuentes costos tanto a nivel personales como de salud pública.

Según Mark L. Braunstein [5] los registros médicos electrónicos pueden clasificarse en tres diferentes tipos, Registros Médicos Electrónicos creados por profesionales, Registros Médicos Personales creados y administrados por pacientes y'El Intercambio de Información Médica la que es usada para compartir información digital entre centros médicos y/o dispositivos personales del paciente. Es así como estos diferentes canales de información pueden ser fuente para crear conocimiento y cuidado de los pacientes.

Como parte de los avances que la comunidad científica ha estado trabajando, está la utilización de DNNs para resolver problemas de visión, clasificación, predicción, entre otros. Es posible apreciar la versatilidad que esta técnica al ser aplicada para resolver problemas de visión en autos autónomos, en cuyo caso el impacto ha sido tal, que el fabricante de tarjetas de video NVIDEA afirma que "Tu próximo auto podría ser tu primer supercomputador"[31] al presentar en la feria CES 2016 un supercomputador diseñado para procesar inteligencia artificial con foco en la industria automotriz [31].

El reconocimiento de voz para dar instrucciones a los asistentes de virtuales de diferentes plataformas como Siri (ios), Cortana (Widnows) o Google Now (Android) su uso es cada vez más común entre las personas. Esto se debe principalmente a cómo estos diferentes motores han mejorado su capacidad para reconocer texto sobre un audio, para lo que también se está utilizando DNNs [18].

1.1.2. Datos Disponibles

Los datos requeridos para entrenar y analizar un modelo predictivo son tan importantes como el modelo en si mismo. Según el dicho 'Garbage In, Garbage Out', que hace referencia a que sin importar la capacidad y Correctitud del modelo de minería de datos, si los datos ingresados no son correctos, no es posible esperar que los resultados lo sean.

Estas inquietudes, son las que en 1980 motivaron a la publicación de la base de datos 'MIT-BIH Arrhythmia Database' la que consta de 48 registros electrocardiográficos con una duración de 30 minutos cada, capturado desde 47 pacientes distintos, realizados entre los años 1975 y 1979. Debido al impacto generado por la base de datos, se realizó una publicación en 2001 por Moody [28] en dónde se detalla la historia de ésta base de datos y cómo se convirtió en un referente de comparación hasta el día de hoy diferentes tecnologías de clasificación aplicadas a la electrocardiografía.

Los principales pasos realizados para la elaboración de esta base de datos están ilustrados en la figura 1.2, que representan a grandes rasgos la metodología y entrega una visión de la cantidad horas hombre requeridas para anotar y clasificar cada latido presente en las 24 horas de registros electrocardiográficos. Siendo esta la principal razón por la que en esta tesis no contempla dentro de sus objetivos la generación y captura de datos propios. Así mismo, al no realizase la captura de electrocardiografías en pacientes chilenos, no se realizarán análisis de resultados con pacientes nacionales.

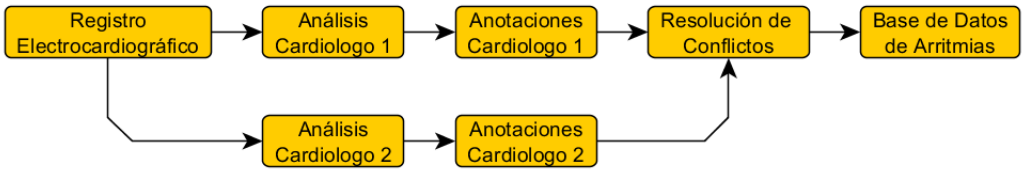

Figura 1.2: Proceso de Registro Electrocardiográfico en la Base de Datos MIT-BIH [28].

Parte del desafío que sostiene un estudio de minería de datos es el obtener un espacio muestral estadísticamente significativo. La primera impresión que genera la base de datos MIT-BIH, con tan solo 48 registros no es posible establecer dicho criterio. No obstante, lo que se espera obtener a partir de dichos registros, es la capacidad de detectar una pulsación cardíaca y de clasificar dicha como normal o anormal. Es así como los 48 registros de 30 minutos entregan un espacio de 109.000 [28] pulsaciones para realizar el estudio. Es desde este punto de vista, que es considerable como suficiente el espacio muestral, para realizar el estudio de minería de datos.

Es importante mencionar que esta base de datos posee un sesgo, al capturar datos sobre pacientes con enfermedades en su ritmo cardíaco, sobre representando la muestra con registros anormales. Sin embargo, el registro de pulsaciones anormales en pacientes enfermos, no es predominante, por que cada elemento del electrocardiograma es vital para el cumplimiento de la función cardíaca, de esta forma si la arritmia está presente de forma continua en la muestra, lo más probable es que el paciente se encuentre en el proceso de manifestar un ataque cardíaco. Este sesgo en la muestra es justificado por que la principal motivación de

la base de datos es la investigación y estudio de modelos de minería de datos, para lo que se requería que la muestra de datos anormales fuese de mayor tamaño a la que sería representada si es que para el estudio se hubiese utilizado una muestra representativa de una población normalmente distribuida.

1.2. Problema u Oportunidad

Actualmente existen múltiples estudios y publicaciones que utilizan técnicas de minería de datos sobre registros electrocardiográficos, con los que se estudian diferentes hipótesis, dentro de las que se encuentra la detección de múltiples enfermedades cardíacas. Al utilizar dichas técnicas diseñadas para la detección temprana de fallas en el corazón que puede prolongar la vida y/o la calidad de vida de los pacientes al suministrar el tratamiento adecuado, al predecir o alertar la presencia de una anomalía.

Estos estudios son realizados de forma autónoma (aislada sin comunicación entre otras investigaciones), por lo que cada investigador no reutiliza el código que otro pudo haber desarrollado, partiendo esencialmente desde cero, existiendo la oportunidad para optimizar el trabajo de los investigadores al realizar un Framework que provea de las herramientas mínimas necesarias para realizar investigación sin tener que incurrir en el costo de desarrollo de estas herramientas.

De esta forma se ha elaborado una lista de requerimientos mínimos, basadas en aspectos técnicos, metodológicos y descritos en publicaciones de diferentes estudios de clasificación de pulsaciones cardíacas, que utilizan técnicas de minería de datos y los propios requerimientos que la técnica de aprendizaje profundo de redes neuronales posee. Es así como se pretende resolver la oportunidad de diseñar un Framework, que facilite la realización de futuros estudios sobre datos electrocardiográficos, como el estudio de la aplicación de metodologías de aprendizaje profundo con redes neuronales para la detección y clasificación de pulsaciones.

1.3. Hipótesis de Investigación

La metodología de minería de datos de aprendizaje profundo, con redes neuronales, es capaz de clasificar y detectar diferentes componentes de un electrocardiograma en tiempo real.

1.4. Objetivos

A continuación se detallan los objetivos generales y específicos, que son desarrollados en los siguientes capítulos, que han sido propuestos para el desarrollo de esta tesis.

1.4.1. Objetivo General

Diseñar, implementar un prototipo de Framework que permite simular la aplicación y entrenamiento, de un modelo de minería de datos de aprendizaje profundo con redes neuronales sobre registros de electrocardiografía.

1.4.2. Objetivos Específicos

1. Identificar los requerimientos de la plataforma, de los modelos y diseñar los diferentes flujos de comunicación necesarios para su funcionamiento (Capítulo 3).

2. Desarrollar un prototipo del Framework, que implemente los flujos mínimos diseñados y cumpla con los requerimientos levantados (Capítulo 4).

3. Estudiar el rendimiento de las redes neuronales profundas tanto para la extracción de componentes principales, como para la detección de enfermedades (Capítulo 5).

1.5. Metodología

Se ha considerado como parte esencial del trabajo la aplicación de buenas prácticas, lo cuál es difícil de probar en si mismo, por el simple requerimiento de que hay que decir por qué las otras prácticas no son mejores y los criterios de elección pueden ser sesgados favoreciendo a una práctica en particular. No obstante, la implementación de buenas prácticas posee como eje principal dar reglas al crecimiento de la aplicación, evitando un crecimiento orgánico y que suele reflejarse en un resultado poco escalable y de difícil mantención. La experiencia muestra que cambios en estructuras y procesos, para la aplicación de buenas prácticas resulta frustrante y en muchos casos más costoso que el beneficio adquirido [30]. Siendo estas las principales razones de la aplicación de buenas practicas, el trabajo posee un foco claro en el estudio del marco necesario para la realización de estudios de minería de datos, de esta forma no se contempla la realización de estudios de impacto que estas poseen, por lo que solamente son utilizadas.

Por la naturaleza dispar de cada uno de los objetivos propuestos, es que la metodología a utilizar varía entre ellos, con fin de lograr un desarrollo confiable. Primero se realiza una revisión del estado del arte, de los diferentes tópicos que abarca la memoria, esta se encuentra detallada en el (Capítulo 2). Siguiendo con el desarrollo de los objetivos específicos, de ellos se desprenden 5 procesos claves, los que se detallan a continuación.

1.5.1. Levantamiento de Requerimientos

La visión de esta tesis, es ser la primera piedra de un macro ecosistema de desarrollo entorno a la aplicación de minería de datos sobre información generada en tiempo real, es

por esto que el levantamiento de requerimientos se realizará entorno al estudio de los diferentes patrones y buenas prácticas.

Para esto se procede con el levantamiento de los servicios y requerimientos mínimos para el funcionamiento del Framework. Siguiendo con el estudio de los diferentes patrones de arquitecturas, considerando que la plataforma es parte de un todo, cuyo objetivo es ser parte de un sistema para la investigación y aplicación de diferentes técnicas de minería de datos.

Es así como desde los requerimientos mínimos serán complementados con los requerimientos de modelo de aprendizaje profundo con redes neuronales, por lo que se espera el desarrollo de un sistema sustentable y extensible.

1.5.2. Procesamiento de Datos

Los datos utilizados en el sistema serán extraídos de una base de datos existente, este echo no representa una garantía. Así mismo, estos deben ser leídos de los archivos fuente y traducidos para su análisis, esto por que la base de datos no es más que una colección de archivos de texto con datos formateados. Es así como parte del desarrollo de la plataforma requiere que una estructura de datos sea modelada y posteriormente poblada con los datos obtenidos de la Base de Datos (DB) MI. Para este proceso de utilizará la metodología CRISP-DM[37].

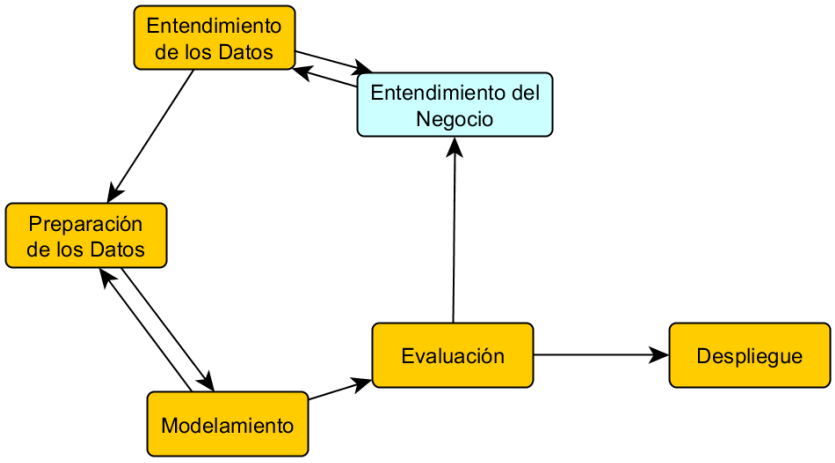

Figura 1.3: Proceso de estudio de datos mediante la utilización de CRISP-DM [37].

Entendimiento del Negocio

Entender el fin ultimo del trabajo y que información se requiere para logar este fin, son los dos grandes temas que se deben evaluar previos a la realización de cualquier estudio de minería de datos. Es así como entender el macro objetivo con el que se desarrolla este trabajo y su fin último es parte de los análisis que se ha expuesto en el presente capítulo introductorio.

Por otro lado, se debe entender que la evaluación comercial del trabajo no es parte del marco en el que se desenvuelve este, no obstante, si lo es la evaluación de impacto, en cuanto al porcentaje de desarrollo requerido, que no es específico al modelo. Este porcentaje representa el potencial de ahorro que el Framework tendría para futuras investigaciones. Esta evaluación será realizada en el Capítulo 5. En este capítulo se implementa un algoritmo de aprendizaje profundo con redes neuronales para la detección y segmentación de latidos cardíacos, utilizando el Framework y finalmente se estudia la cantidad de código reutilizado en comparación con el código requerido para el desarrollo de cada uno de los modelos, considerando que el tiempo requerido para el desarrollo del código que pertenece al modelo, es el mismo que para el desarrollo de los modelos se concluye con un análisis económico que Framework implica en términos de horas hombre del desarrollo.

Entendimiento de los Datos

Entender la naturaleza del electrocardiograma, cómo se captura, el origen de la señal electrocardiográfica y el patrón que sigue una enfermedad cardíaca, son los principales elementos del entendimiento de los datos. Estos elementos serán estudiados principalmente en el Capítulo 2, para posteriormente estudiar un modelo de datos en el Capítulo 4.

Preparación de los Datos

La preparación de los datos cubre todas las actividades que involucran la construcción del set de datos final. Este proceso comienza con la lectura de los archivos brutos que componen la base de datos de arritmias, limpieza de estos, creación del modelo de datos, extracción de los componentes principales y la posterior población de la base de datos. Estas actividades cubren parte del Capítulo 2, pero mayoritariamente son desarrolladas en el (Capítulo 4).

Modelamiento

Tal como existen múltiples formas de modelar un problema de minería de datos, existen múltiples técnicas para aplicar estos modelos a los datos. De esta forma el problema de detección de arritmias serán expuestas diferentes formas de modelar el problema en la Subsección 2.1.4. Como el foco del trabajo no es la realización de una evaluación cualitativa de los diferentes modelos, será esto estudiado comparando los resultados de especificidad y sensibilidad obtenidos por otros estudios.

Los principales componentes del modelamiento son, la selección de la técnica y la construcción del modelo, ambos no serán discutidas en profundidad en el trabajo. Debido a que el trabajo es él estudio de una técnica y su rendimiento en el contexto de la electrocardiografía.

Evaluación

Parte del desarrollo del Framework es la capacidad de poder evaluar y comparar dos modelos que posean el mismo objetivo a predecir o clasificar. De esta forma será parte del diseño la implementación de un set de herramientas estadísticas que permitan analizar el modelo sin interferir en el desarrollo de éste, lo que será detallado en el Sección 5.3.

Implementación

La etapa final de metodología, consta de 4 elementos, planes de implementación, monitoreo y mantenimiento, producto final y revisión final del proyecto. Estas etapas son desarrolladas en el Sección 4.6.

1.5.3. Modelo de Detección de Arritmias

Existen múltiples ejemplos en la literatura que modelan la detección del complejo QRS y de arritmias, las diferentes combinaciones de estos modelos han sidos evaluados a lo largo de la últimas dos décadas con la intención de encontrar mejores rendimientos en términos de hardware, compresión de datos y capacidad de predicción. Estos modelos son detallados y comparados con los resultados obtenidos del modelo implementado en Sección 5.4.

Esencialmente los modelos relacionados con la electrocardiografía y la detección del complejo QRS (ver Figura 2.1) esencialmente son evaluados utilizando dos parámetros, la sensibilidad y especificidad positiva [25].

$$Sensibilidad = \frac{VP}{VP + FN} \tag{1.1}$$

$$Especificidad^{+} = \frac{VP}{VP + FP} \tag{1.2}$$

Dónde VP es la cantidad de Verdaderos Positivos detectados, FN los Falsos Negativos, FP los Falsos Positivos.

Los resultados obtenidos por una investigación pueden ser clasificados en 3 categorías según [25]:

- Resultado confiable, es aquel en que el algoritmo fue probado en una base de datos estándar.
- Resultados poco confiables, es aquel en el que algoritmo fue probado en partes de una base de datos estándar.
- Resultados no confiables, es aquel resultado en que el algoritmo fue probado en una base de datos privada o no estándar.

A su vez se describen otros elementos como el no publicar las ecuaciones y/o pseudo código, como factores que catalogan a una investigación de poco confiable. Por estas razones es que la investigación se encuentra documentada y tal como se menciona en la Subsección 1.1.2 los datos utilizados provienen de una base de datos estándar.

1.5.4. Desarrollo del Framework

El desarrollo del Framework requiere el levantamiento de requerimientos, diseño y elaboración. Sin olvidar el fin último del proyecto, el cuál es la creación de un ambiente para la investigación de técnicas de minería de datos sobre el electrocardiograma, se utilizará una metodología ágil para el desarrollo de éste.

La metodología ágil posee como pilares el desarrollo iterativo y colaborativo. De esta forma se entiende la necesidad de que más de un programador este trabajando en el mismo proyecto, como también que el desarrollo debe ser adaptativo y que recibe reglamentación de los resultados a medida que se va iterando en el proceso [9].

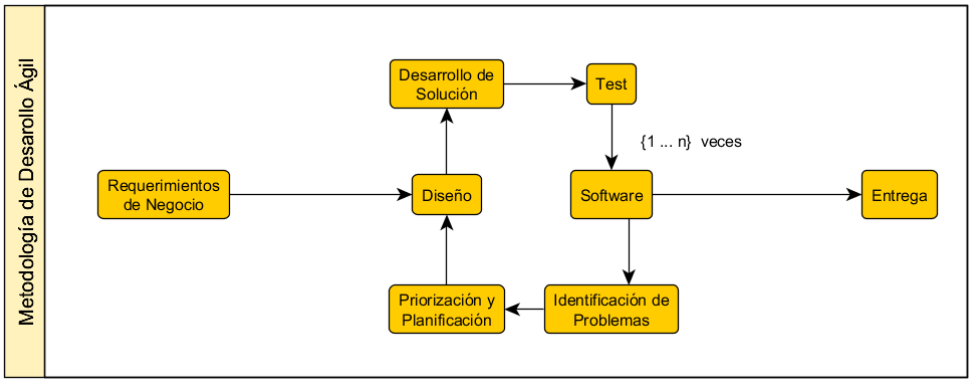

Figura 1.4: Flujo de trabajo en la metodología ágil de desarrollo.

Ésta metodología esta compuesta de dos etapas y un ciclo el que puede visualizarse en la figura 1.4. El proceso comienza con el levantamiento de los requerimientos de negocio, los que son listados y a continuación se les asigna una prioridad, para posteriormente ir uno a uno ingresando los requerimientos al ciclo de desarrollo. Este ciclo consta de 6 etapas, las que se diseñan, desarrollan, prueban y se retroalimentan de los resultados del desarrollo, de esta forma se puede tanto reorganizar las prioridades de los requerimientos, como crear unos nuevos según lo aprendido.

Finalmente el proceso culmina cuando todos los requerimientos son desarrollados o solucionados, en ese momento el código que fue pasando por el ciclo de desarrollo múltiples veces, y creciendo en cada iteración, es finalmente entregado.

1.6. Resultados Esperados

Los principales resultados esperados son:

1. Estudio del modelo de procesos de negocios para el desarrollo de una investigación en minería de datos sobre señales electrocardiográficas.

2. Reglas para el desarrollo adaptativo y colaborativo del programa prototipo, teniendo en cuenta que éste posteriormente pueda ser complementado y mejorado para ser llevado a producción.

3. Características principales de la primera derivación electrocardiográfica.

4. Set de reglas: que permite detectar un latido incompleto.

5. Set de reglas: que permite detectar los componentes principales de un latido.

6. Set de reglas: que permite caracterizar un patrón de arritmia.

7. Script de minería de datos, que detecte arritmias en registros electrocardiográficos y utilice el Framework piloto programado.

1.7. Contribución de la Memoria

Al final de la memoria, ésta contribuirá con:

1. Framework especializado en la aplicación de minería de datos sobre registros electrocardiográficos.

2. Implementación de un modelo de DNNs entrenadas para la detección de arritmias.

3. Estudio que compara el rendimiento del modelo implementado, con los resultados obtenidos por otras investigaciones.

Capítulo 2

Marco Conceptual

2.1. Electrocardiograma (ECG)

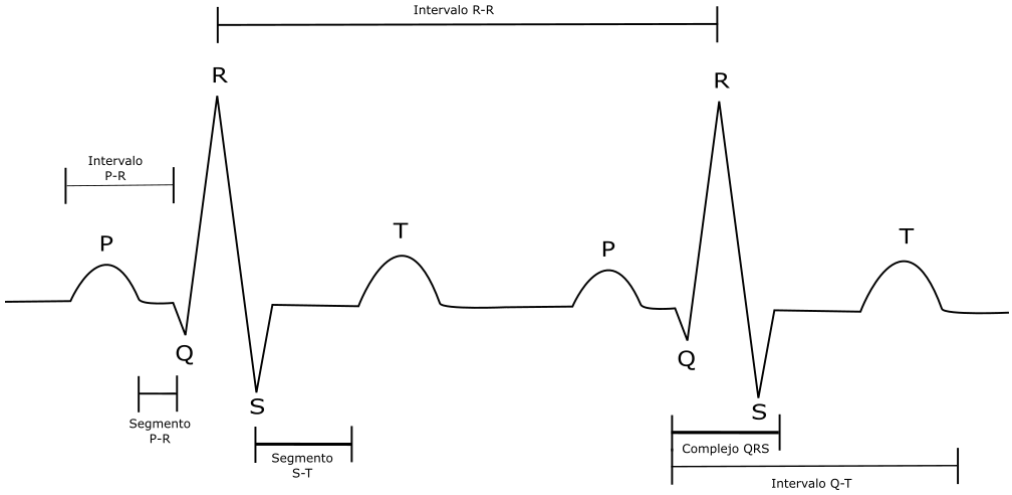

Figura 2.1: Esquema representativo de un electrocardiograma en condiciones normales y sus principales segmentos e intervalos [13, 21].

El electrocardiograma evaluá el ritmo y conducción del movimiento cardíaco mediante la examinación de la duración, amplitud y forma de la diferencia de potencial eléctrico que produce el movimiento de iones para la despolarización auricular (onda P), despolarización ventricular (complejo QRS) y la repolarización ventricular (onda T).

El electrocardiograma posee una naturaleza oscilatoria y periódica, el ciclo consiste en una serie de ondas denominadas P-QRS-T, asignándose la letra al valor máximo o mínimo de la onda según corresponda, cada una de estas ondas posee características que permiten realizar un clasificación, en dónde se destacan la amplitud y duración. El ECG provee información valiosa sobre el funcionamiento del corazón, el primer registro electrocardiográfico fue realizado por el fisiólogo inglés August Waller en 1887, no obstante W. Einthovem es llamado el padre de la electrocardiografía, realizando su primer registro en 1902 y recibiendo el premio Nobel por su trabajo en 1924.

En el normal funcionamiento interno del corazón se ven envueltos principalmente 3 procesos: químico, eléctrico y mecánico. Adicionalmente de forma externa influyen el sistema circulatorio y el nervioso. De este último los nervios simpático y parasimpático controlan de forma externa la frecuencia cardíaca. El nervio simpático puede aumentar el volumen de la sangre bombeada (salida cardíaca) en 100 veces los niveles de reposo, por otro lado el nervio parasimpático posee la capacidad de disminuir la frecuencia hasta a niveles cercanos a cero [16].

Sin embargo como menciona Rajarshi Gupta [15] *"la forma y magnitud del ECG varia ampliamente entre la población de diferentes continentes, y es determinada por hábitos alimenticios y factores hereditarios demográficos"*. Este es uno de los motivos por lo que los algoritmos no pueden realizar clasificaciones con un 100 % de precisión, sin embargo esta misma razón es el origen de nuevas investigaciones.

2.1.1. El Corazón

El corazón es él órgano encargado de la circulación sanguínea a través del sistema circulatorio. Éste consta de dos aurículas y dos ventrículos, los que conforman dos bombas separadas, cuyo funcionamiento es simultáneo. Ambas, bombean sangre en dos sistemas diferentes de circulación (Menor y Mayor), los que transportan sangre al sistema pulmonar y general, respectivamente (ver figura 2.2).

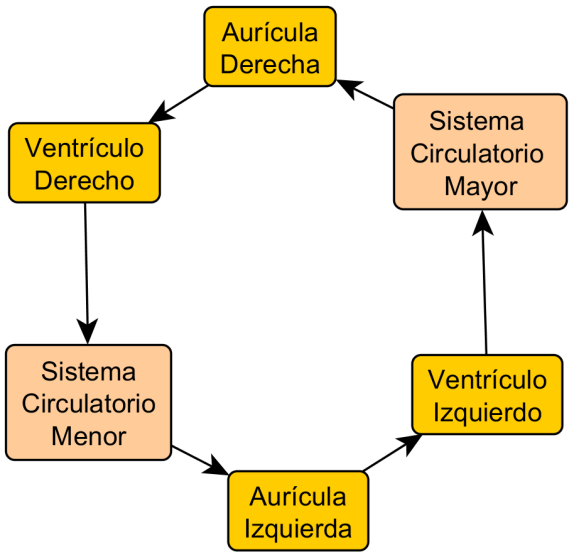

Figura 2.2: Flujo Sanguíneo a través del Sistema Circulatorio [23].

La acción de bombeo que el corazón realiza, se lleva a cabo por la contracción de las aurículas y ventrículos, contrayéndose primero las aurículas y luego los ventrículos. Estas contracciones son realizadas gracias a la estimulación eléctrica de los músculos cardíacos, la que es ejecutada mediante un impulso eléctrico que es transmitido por las diferentes estructuras nerviosas que posee el corazón. Esta estimulación eléctrica consta de dos etapas,

la primera llamada despolarización que induce contracción muscular, posteriormente el nervio debe volver a su estado basal mediante el proceso llamado repolarización, mientras esto sucede el corazón se relaja para poder volver a iniciar el ciclo.

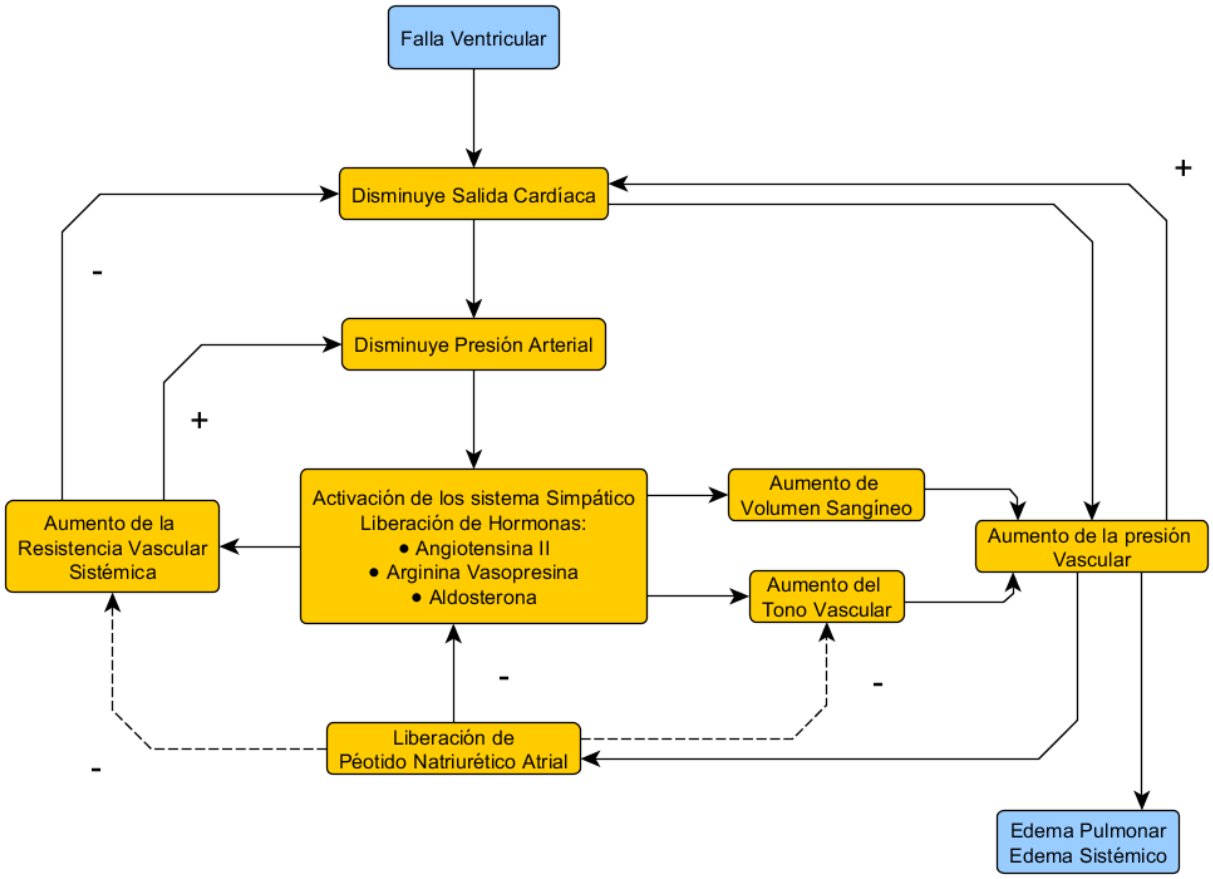

Figura 2.3: Proceso de cambios asociados a una falla ventricular, el que dispara múltiples mecanismos tanto nerviosos como hormonales, con el único objetivo de compensar la falla cardíaca y así poder entregar parte de la función cardíaca. Parte esencial del proceso es el equilibrio que se espera alcanzar posterior a la activación del proceso, no obstante si la falla no es compensada, el proceso no llegará a un equilibrio y puede provocar tanto un edema pulmonar como sistémico [23]

La velocidad de transmisión eléctrica no es controlada, debido a que ésta es una reacción química que se desplaza a través de las estructuras nerviosas del corazón. Sin embargo el corazón realiza una pausa posterior a la contracción auricular para llenar los ventrículos y luego finalmente contraer éstos. Este fenómeno se da gracias a una estructura llama nódulo auriculoventricular, el que transmite el impulso eléctrico de una forma más lenta, para posteriormente volver a ser transmitido rápidamente por el Haz de Hiz [10], de esta forma la pausa que es posible apreciar entre las contracciones auriculares y ventriculares, no se debe a que el impulso se detenga, sino a una transmisión lenta entre estructuras que están unicamente conectadas eléctricamente por el nódulo mencionado.

Por lo anteriormente descrito es importante mencionar que el proceso de contracción cardíaco ocurre de forma automática una vez alcanzado el umbral de activación en el nódulo

sinusal, por esa razón es que la frecuencia cardíaca es principalmente controlada mediante la variación de las pausas entre pulsaciones.

2.1.2. Captura del Electrocardiograma

La electrocardiografía es la medición de la actividad eléctrica del corazón, la que es posible medir gracias a el campo magnético generado por el desplazamiento de cargas eléctricas realizado por los procesos de polarización y despolarización. Estas mediciones son clasificadas en 2 tipos:

1. Según su Naturaleza:
   (a) Exámenes Rutinarios.
   (b) Exámenes Especializados.
2. Según su Metodología:
   (a) **Invasivos** Capturados desde las paredes del corazón.
   (b) **Semi-Invasivos** Registrados desde el esófago.
   (c) **No Invasivos** Registrados desde la superficie de la piel, principalmente en el pecho.

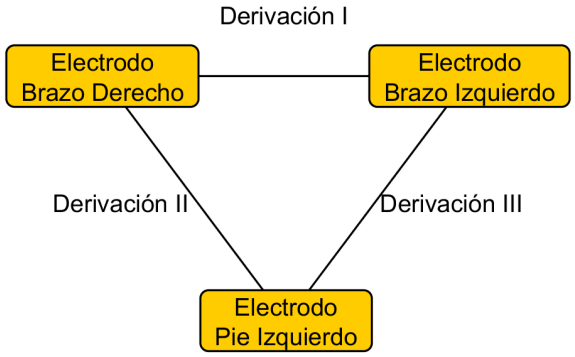

Figura 2.4: Derivaciones generadas por cada par de electrodos [10]

El desplazamiento de las cargas eléctricas induce un campo magnético que es posible medir desde la superficie de la piel, este es expresado en una diferencia de potencial eléctrico del orden los mili volt. En consecuencia, el ECG es simplemente la medición del voltage entre dos puntos de la superficie de la piel, pero depende de la distribución de los Electrodos en el cuerpo, porque cada par de electrodos captura un plano electromagnético, el que refleja una parte de la actividad del corazón. De esta forma es como a cada uno de estos planos electromagnéticos se les denomina derivación, como es posible ver en la figura 2.4. La derivación estándar es la llamada AVF [10], en dónde se utiliza el electrodo del pie izquierdo como

positivo y tanto el electrodo de la mano izquierda como el de la derecha como negativos.

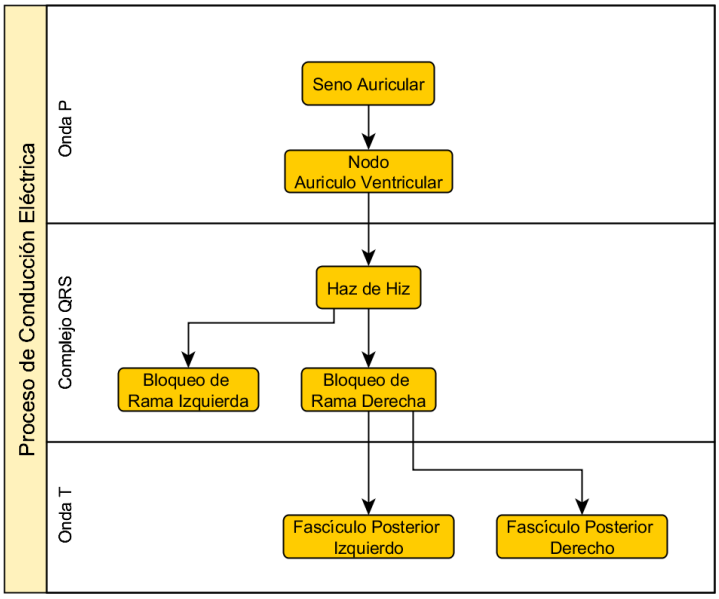

Figura 2.5: Proceso de Conducción Eléctrica segmentado por etapa visible en el electrocardiograma [10].

2.1.3. Arritmia

La arritmia es una mala función cardíaca, que resulta de la ejecución defectuosa del ritmo cardíaco, esta se puede clasificar en grupos, taquicardia, bradicardia, bloqueos, contracciones prematuras y Desórdenes de repolarización (LQT).

La taquicardia es un desorden de ritmo que afecta la frecuencia cardíaca, al igual que la bradicardia, estos desordenes se caracterizan por la realización de un proceso normal de contracción cardíaca, pero su frecuencia se encuentra seriamente afectada. En el caso de la taquicardia, se define con frecuencias superiores a las 100 pulsaciones por minuto en pacientes en reposo, por el contrario la bradicardia, es el reflejo de un ritmo cardíaco inferior a las 60 pulsaciones por minuto en reposo [16].

Los bloqueos cardíacos, pueden existir en cada una de las estructuras transmisoras del impulso eléctrico que provocan las contracciones tanto ventricular como auricular. De esta misma forma los bloqueos pueden ser a su vez parciales, impidiendo de forma arbitraria el paso de la señal eléctrica (ver figura 2.5) , un ejemplo de esto es el bloqueo incompleto atrioventricular, el que genera razones de contracción de la forma 2:1, 3:2 y 3:1 [23], en dónde por ejemplo las aurículas se contraen dos veces por cada vez que lo hacen los ventrículos, efecto visible en el ECG por la presencia de dos ondas P consecutivas.

Las contracciones prematuras de las diferentes estructuras, puede ser originada por cicatrices, placas calcificadas, irritación, drogas, nicotina, cafeína e incluso estimulaciones mecánicas entre otros. La principal evidencia de contracciones prematuras ECG es la malformación de

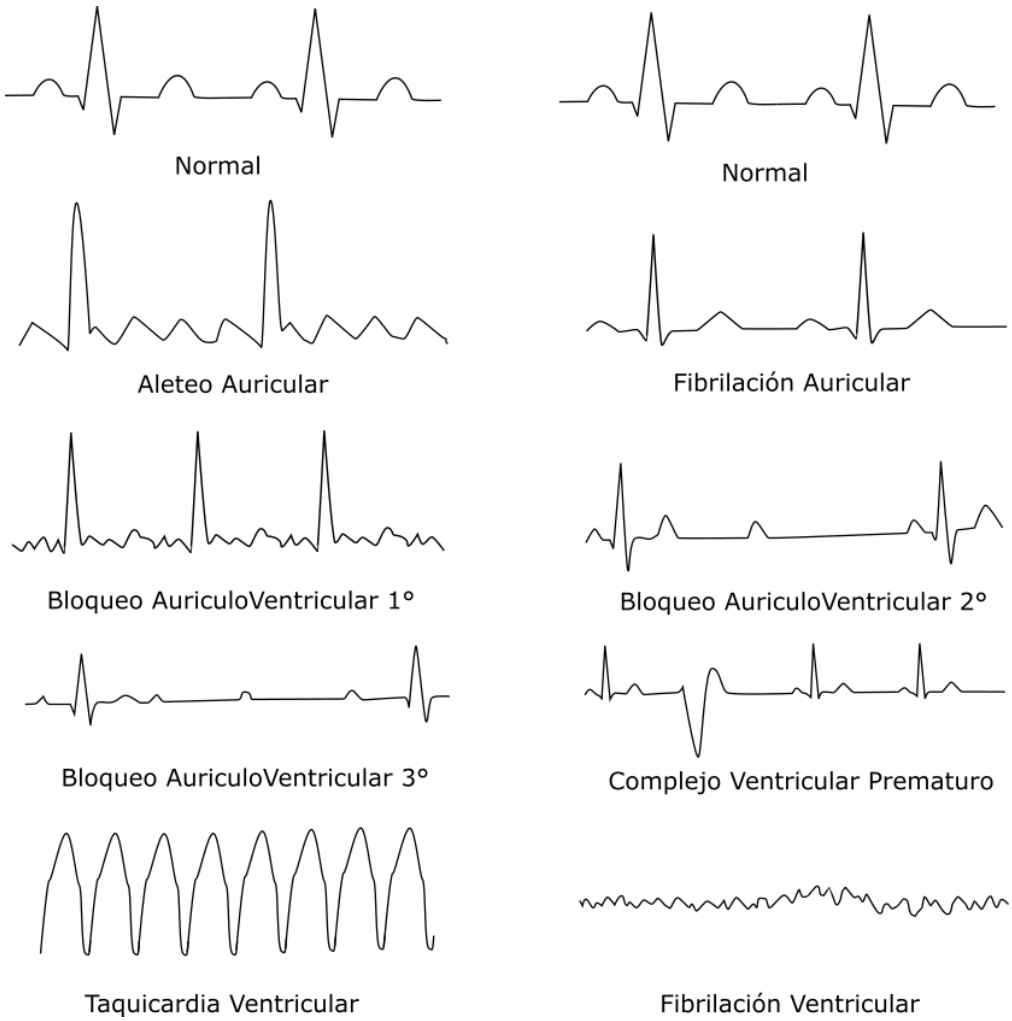

Figura 2.6: Ejemplos de diferentes tipos de arritmias, fuente [23]

las estructuras o reducción de los intervalos normales, por ejemplo entre P-R, prolongación del complejo QRS-T o un aumento en el voltaje QRS [23].

De las ejecuciones cardíacas erroneas descritas anteriormente, la más grave de todas son los LQT por que dentro de sus efectos esta la taquicardia y la fibrilación. Ésta es generada por una malfunción del proceso eléctrico de polarización - despolarización.

La fibrilación ventricular es una mala función eléctrica de los ventrículos del corazón, que contraen porciones del miocardio de forma no coordinada, esta es la razón por la cual el corazón no bombea sangre. Esta condición si no es detenida entre 1 a 3 minutos es irremediablemente letal y una vez manifestada el paciente solo posee entre 3 a 5 segundos de consciencia, por la falta de sangre oxigenada en el cerebro [23].

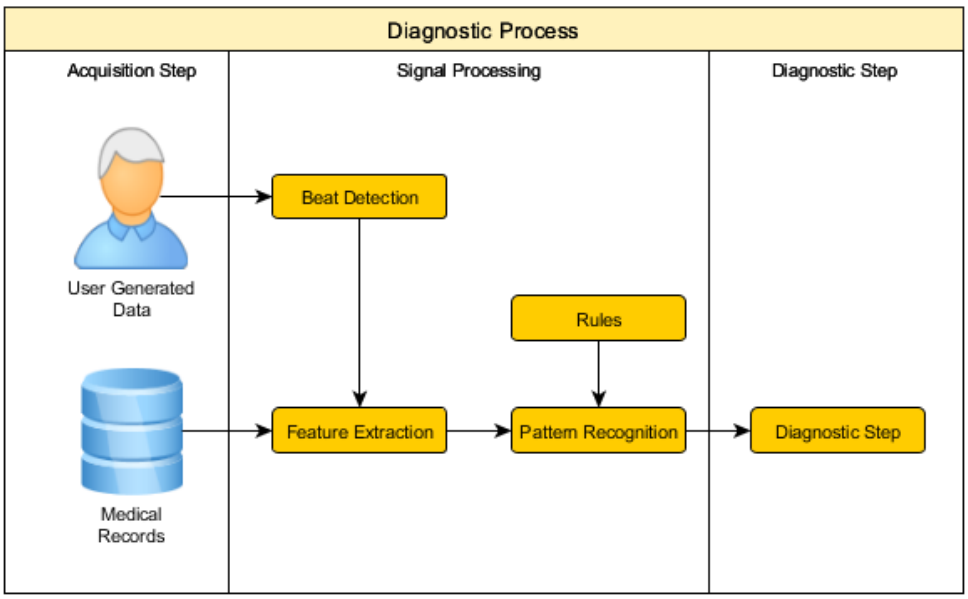

Figura 2.7: Proceso clásico de diagnóstico automático [13].

2.1.4. Métodos para el Análisis de Señales Electrocardiográficas

Son diversas las metodologías y técnicas que han sido utilizadas para detectar anomalías en el electrocardiograma, sin embargo todas poseen una componente común, éste es la necesidad de extraer de formá rápida y precisa los componentes principales del electrocardiograma. Posteriormente se utiliza la reducción de dimensión para ajustar un modelo predictivo, lo que corresponde las fases de reconocimiento de patrones y diagnóstico en la Figura 2.7.

Minería de Datos

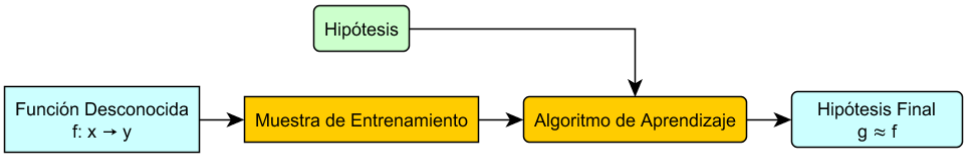

Figura 2.8: Estructura básica de un problema de aprendizaje automático (Rehacer Figura) fuente [3]

Minería de Datos es el término que describe el proceso de extracción de información valiosa desde una fuente rica en datos, la que es utilizada para responder preguntas que la información atómica no es capaz.

La idea es traspasar criterios cualitativos en la señal primitiva de electrocardiografía a un criterio cuantitativo, que pueda ser fácilmente clasificable por un computador. Pero para esto primero hay dos preguntas que se deben responder:

- ¿Qué es lo que se quiere predecir?

  La presencia de arritmia en una señal electrocardiográfica.

- ¿Con qué datos se quiere predecir?

  Utilizando la primera derivación del ECG, el cual esta directamente relacionada con el funcionamiento del corazón.

Cómo es mencionado en múltiples publicaciones [14, 13] el proceso de diagnóstico automático puede ser generalizo en 3 etapas independientes, detección del latido, extracción y segmentación de características principales y clasificación. Este proceso esta limitado por la capacidad de ingeniería para diseñar y modelar un extractor de componentes principales que procese los datos brutos [26].

Extracción de Componentes Principales

La extracción de características principales en términos generales es una reducción dimensional del problema o interrogante que se quiere resolver, sin perder la información relevante del problema en cuestión. En el caso propuesto en esta tesis históricamente se han utilizado diferentes técnicas para encontrar los segmentos e intervalos descritos en la Figura 2.1, posteriormente utilizando éstos valores se procede a estudiar modelos de clasificación que puedan utilizar dichos valores.

El proceso clásico de extracción de los componentes principales comienza con la detección del complejo QRS. Luego se deben identifican los máximos de las diferentes ondas (PQST), para lo que se utiliza su posición relativa con el complejo QRS y con estos los segmentos e intervalos [27].

Es así como múltiples métodos han sido propuestos para detectar el complejo QRS, como los derivados de algoritmos, Redes Neuronales (NNs), transformada de Wavelet, técnicas de filtrado, transformada de Hilbert, modelos markovianos, entre otros [25].

Redes Neuronales (NNs)

Las NNs típicas son una red interconectada de neuronas, también llamadas nodos o unidades. Éstas operan de forma síncrona es decir sólo reaccionan o entregan una salida al recibir un estímulo o entrada de datos, siendo éste un elemento diferenciador, respecto a una red neuronal biológica. Tanto la entrada como la salida de una red neuronal debe ser un número de punto flotante.

La entrada de datos o estímulo, se realiza en la capa de entrada (Input Layer), posteriormente las neuronas de esta capa entregan el resultado de la evaluación de su función de activación a la capa oculta (Hidden Layer) de las que pueden existir más de una capa oculta y finalmente la capa de salida (Output Layer) ver Figura 2.9.

Figura 2.9: Macro proceso de una red neuronal.

Todas las NNs comparten componentes característicos, como neuronas, ponderadores, funciones de activación y capas de neuronas. Las neuronas de una misma capa comparten la misma función de activación, pero no todas las estructuras poseen capas para organizar las neuronas. En la función de activación de una neurona se suman todos los valores de entradas x_{i} multiplicados por el ponderador w_{i}, finalmente la suma de estos valores ponderados se evalúa en la función de activación Φ siendo el valor resultante de la evaluación de la función de activación el estímulo o entrada enviada a las neuronas de la capa superior o de la capa de salida.

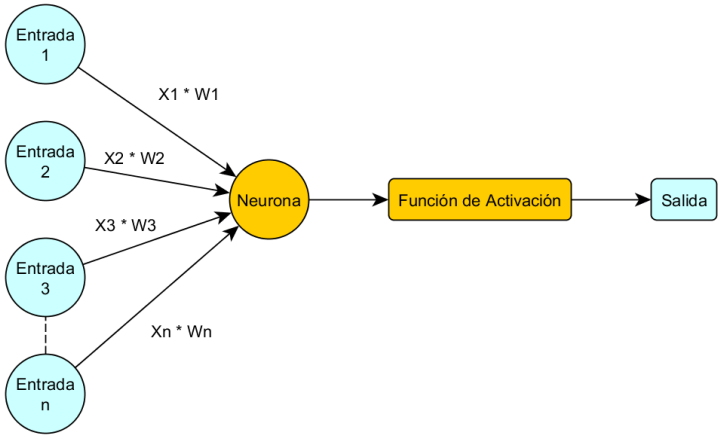

Figura 2.10: Funcionamiento básico de una neurona.

$$f(x_{\mathrm{i}}, w_{\mathrm{i}}) = \Phi(\Sigma_{\mathrm{i}\in I}(w_{\mathrm{i}} * x_{\mathrm{i}})) \tag{2.1}$$

Las funciones de activación tradicionalmente son linear, binaria, sigmoidea, tangente hiperbólica, rectificación de unidad linear y softmax.

Linear

$$\Phi(x) = x \tag{2.2}$$

Binaria

$$\Phi(x) == \begin{cases} x^2 & : x < 0 \\ x^3 & : x \geq 0 \end{cases} \tag{2.3}$$

Sigmoídea

$$\Phi(x) = \frac{1}{1 + \mathrm{e}^{-}x} \tag{2.4}$$

Tangente Hiperbólica

$$\Phi(x) = \tanh x \tag{2.5}$$

Rectificación de Unidad Linear

$$\Phi(x) = \text{máx}(0, x) \tag{2.6}$$

Rectificación de Unidad Linear

$$\Phi_{\text{i}} = \frac{\text{e}^{z_{\text{i}}}}{\sum_{j \in capa} \text{e}^{zj}} \tag{2.7}$$

Esta última función de activación es de uso común en en la capa de salida y en redes neuronales de clasificación debido a que la naturaleza de ésta función es de relacionar a la neurona con el valor de entrada más alto como miembro de su clase. De esta forma la utilización de softmax obliga a interpretar la salida de la NNs como una probabilidad de pertenecer a una clase específica.

Tradicionalmente existen cinco tipos de neuronas:

1. **Entrada:** neuronas que reciben la entrada de datos.
2. **Salida:** neuronas que comunican o entrgan el resultado final de la red neuronal.
3. **Ocultas** neuronas que reciben y entregan datos exclusivamente desde y hacia otras neuronas.
4. **De Polarización:** son aquellas que siempre producen el valor 1, sin depender de su conexión a la res, éste tipo de neurona es ampliamente utilizadas para ayudar a la red a aprender un determinado patrón.
5. **De Contexto** usadas en Redes Neuronales Recurrentes (RNNs), le permiten a la red tener estados, de esta forma para un mismo estímulo, la red puede entregar dos resultados diferentes debido a las diferencias de estado presentes.

Redes Neuronales de Aprendizaje Profundo (DNNs)

Según Yann LeCun, las DNNs son la tercera generación de investigaciones en inteligencia artificial [26]. Las que han mejorado drásticamente el estado del arte en el reconocimiento del habla, reconocimiento visual de objetos, detección de objetos y aplicaciones en otras áreas como la biotecnología, en el comportamiento de drogas, procesamiento del genoma y reconstrucción de circuitos cerebrales [26].

Los métodos de aprendizaje profundo le permiten a una máquina encontrar múltiples representaciones, en diferentes niveles de procesamiento. Con la composición de estos niveles no lineales de procesamiento, ésta la máquina puede aprender complejas representaciones sin mayor intervención humana [26].

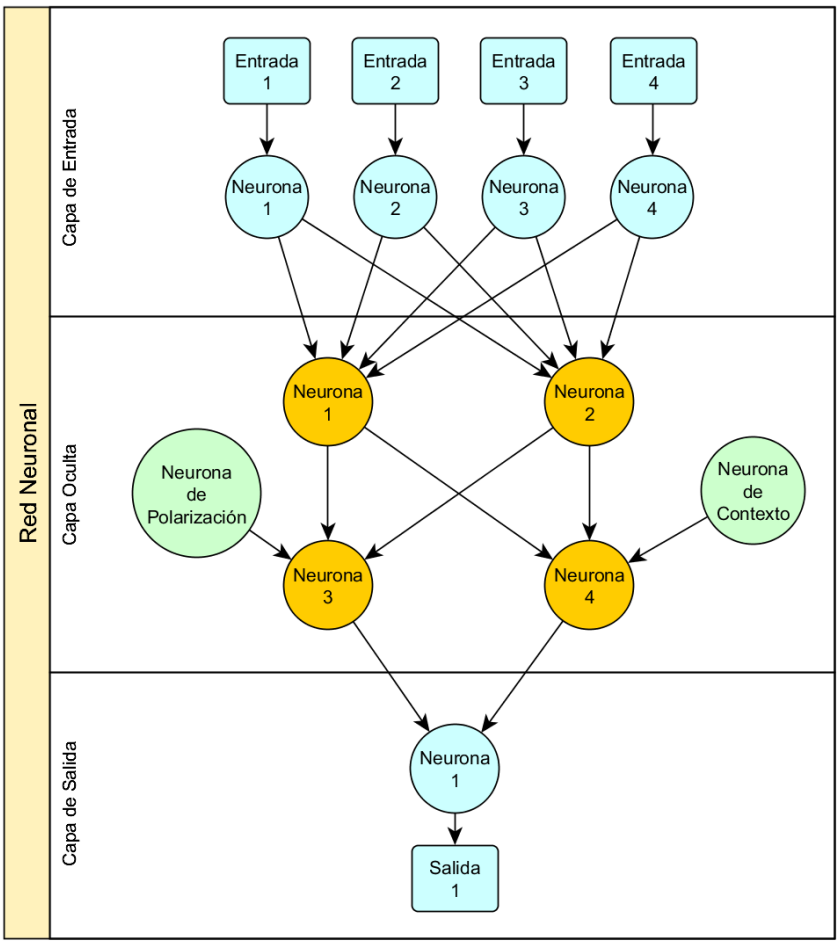

Figura 2.11: Red Neuronal Básica con todos sus componentes.

Otra aplicación en la que las NNs han mejorado el estado del arte, es el trabajo presentado por Alexander Toshev y Chistian Szeged [36]. Quienes propusieron la utilización de DNNs para la estimación de la pose humana a partir de imágenes. Éste problema, es particularmente complejo debido a las pequeñas y en muchos casos, poco visible, que es la posición de las articulaciones, sin considerar los casos en que se debe utilizar un razonamiento holístico para estimar la posición de una articulación, porque ésta simplemente no se ve.

Por definición una red neuronal con dos o más capas ocultas puede clasificarse como de aprendizaje profundo, cuyos componentes claves son [26]:

1. **Datos parcialmente Anotados:** es una metodología en la que se entrena a la red neuronal por etapas. En la primera etapa, se entrena a la red para reconocer el objeto que se busca clasificar (parcialmente anotado), una vez que la red logra reconocer el objeto, se añade más información y se entrena a la red de forma más profunda, para que clasifique los objetos reconocidos.

2. **Rectificador linear de unidades:** es una función de activación utilizada en las capas ocultas, especialmente en Restricted Boltzmann Machines (RBMs) y de forma estándar en Deep Belief Neural Networks (DBNNs).

3. **Redes Neuronales Convolucionales (CNNs):** la utilización del operador matemático en una red neuronal, ha sido clave en las mejoras en el rendimiento y capacidad que las redes neuronales han demostrado en visión computacional. Esto se debe principalmente a la capacidad los filtro de convolución tienen para reconocer patrones y ejes al ser aplicados sobre imágenes, sin embargo, la utilización de éste tipo de filtros en la red neuronal dificulta el entrenamiento de la red por los componentes de los filtros convolutivos, los que son : el número de filtros, el tamaño del filtro, paso, relleno, función de activación, zonas de color y otros elementos visuales [26].

4. **Abandono Neuronal:** consiste en la eliminación de neuronas de la red durante el entrenamiento de la misma, ver Figura 2.12. Éste abandono permite disminuir el sobreajuste de la red y mejorar el rendimiento de la ejecución y del entrenamiento de la red, el que consiste básicamente en la eliminación de aleatoria de neuronas de la red y posteriormente se evalúa el resultado de dicha eliminación [34].

Sin embargo la principal característica de una DNNs es la forma en que ésta es entrenada, por que no solo se entran en términos de una red neuronal tradicional seleccionando los ponderados de cada una de las neuronas de la red, sino que el modelo mismo de cada capa-neurona es auto seleccionado. Con este fin existen factores y ténicas que deben ser considerados. Los factores son el número de capas ocultas, orden de la red, si es convolucional, se utilizarán capas con factores comunes y la estructura de estos factores (ver Figura 2.12), si se utilizará abandono neuronal, tipos de función de activación, entre otros.

Finalmente durante el entrenamiento la selección y valoración de ponderadores puede realizase utilizando diversas metodologías, las más destacadas son: búsqueda en grilla, búsqueda aleatoria, optimización de colonia de hormigas, algoritmos genéticos, programación genética, escalada de hill, Nelder-Mead, Particle Swarm Optimization (PSO) y simulación recocida. Para ver las fortalezas de las diferentes redes ir a Anexos Tabla B.3.

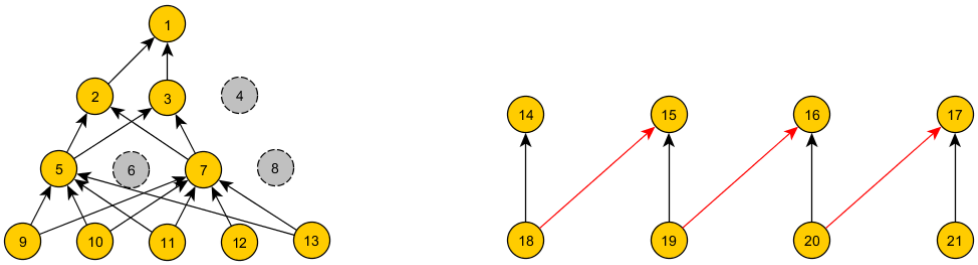

Figura 2.12: Izquierda: red neuronal en la que se han abandonado los nodos 4, 6 y 8, fuente [34]. **Derecha:** ilustración de una transición de capa que utiliza factores compartidos, en esta capa los pesos de las conexiones de color negro son iguales, de igual forma los pesos de las conexiones rojas, fuente [26].

Capítulo 3

Identificación de Requerimientos y Diseño de la Plataforma

3.1. Requerimientos

El diseño del prototipo esta compuesto de dos grandes sistemas que coexisten y se comunican, estos son el sistema de administración web, el que esta basado en la nube y consta de elementos como el sistema de administración y servicios de datos que son prestados al segundo sistema, el Framework. En el presente capítulo se estudiarán los componentes de la plataforma y sus requerimientos de funcionamiento.

3.1.1. Requerimientos de la Plataforma

Se han identificado múltiples fuentes de requerimientos, de las que se ha decidido utilizar solo las que permiten en un conjunto la realización de un prototipo de platafroma que permita estudiar un modelo. No obstante ciertos requerimientos no serán satisfechos por el prototipo, dado que no son críticos para el desarrollo de un modelo, sin embargo si lo podrían ser para algún componente clave de negocio, un ejemplo de esto es la creación de un sistema de registro de usuarios y procesamiento de cuotas de uso, ambas características claves para el negocio pero que no se relacionan con el proceso de análisis de modelos concepto conocido como Acoplamiento.

La principal fuente de requerimientos mínimos, es el propio proceso requerido para la construcción, entrenamiento, prueba y validación de un modelo que utiliza un algoritmo de aprendizaje. Este proceso nace con un conjunto de hipótesis candidatas, para llegar a una hipótesis final (ver figura 2.8).

De los requerimientos expuestos en la Tabla 3.1 se desprenden otros, los que serán detallados en las siguiente secciones.

Fase	Requerimiento
1 Construcción	1.1 Herramientas de visualización (Gráficos). 1.2 Capacidad de realizar Debug e del código del modelo.
2 Entrenamiento	2.1 Ser capaz de implementar aprendizajes por refuerzo, supervisado y no supervisado. 2.2 Configurar criterios de selección de datos para el entrenamiento del modelo. 2.3 Gráfico de componentes principales. 2.4 Gráfico Q.
3 Prueba	3.1 Configurar criterios de selección de datos para la prueba del modelo. 3.2 Herramientas estadísticas, como el cálculo del error, análisis de especificidad, análisis de sensibilidad, tiempo de respuesta.
4 Validación	4.1 Configurar criterios de selección de datos para las pruebas. 4.2 Herramientas estadísticas, como el cálculo del error, análisis de especificidad, análisis de sensibilidad, tiempo de respuesta. 4.3 Gráfico de la curva ROC. 4.4 Validación Cruzada.

Tabla 3.1: Requerimientos para el desarrollo de un modelo de minería de datos.

3.2. Diseño de la Plataforma

La arquitectura de la plataforma consta de 4 elementos, los usuarios, el Framework, la aplicación web y las fuentes de datos, los que son parte del esquema usual utilizado para el reconocimiento de patrones en el ECG (Figura 2.7). Las interacciones de estos elementos y sus componentes (Figura A.1) son la forma en que los requerimientos se propagan por el sistema, formando así el marco completo de requerimientos mínimos.

Parte importante de la arquitectura visible en la Figura A.1 son las capas de la aplicación web (presentación, servicio, negocio, datos y transversales), no obstante, gran parte de estas no son exclusivas del sistema que se está desarrollando, esta razón impulsa la utilización del Framework Spring para el desarrollo de aplicaciones empresariales, que esta licenciado bajo la licencia Apache versión 2.0, el que provee las herramientas necesarias para el fácil desarrollo de la arquitectura web del problema que se esta solucionando.

Como gran parte del diseño de la aplicación web es parte del Framework Spring, solo se profundizará en la descripción de los requerimientos de la capa de presentación, fuente de datos y el motor del Framework, ya que el desarrollo del resto de los componentes será visto en el Capítulo 4.

3.2.1. Capa de Presentación Web

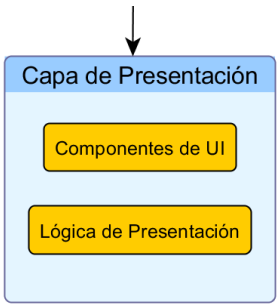

Figura 3.1: Componentes capa presentación.

La capa de presentación es la forma gráfica en que el Framework interactuá con el investigador, entregando herramientas para ayudar a la realización investigación, como su análisis y validación.

Interfaz de Usuario	Requerimiento
5 Resultados (User Interface (UI))	5.1 Herramientas de visualización (Gráficos). 5.2 Análisis de sensibilidad y especificidad. 5.3 Curva ROC. 5.4 Gráfico de Características. 5.5 Gráfico Q. 5.6 Porcentaje de avance, respecto a la muestra seleccionada. 5.7 Ver gráficos transmitidos en tiempo real. 5.8 Desplegar anotaciones (clasificaciones) procesados por el modelo y los propios del modelo.
6 Administración (Lógica)	6.1 Operaciones CRUD sobre resultados. 6.2 Ingreso con usuarios y contraseña (seguridad). 6.3 Conexión al sistema usando una llave de acceso (Token) para el Framework.
7 Configurar (Lógica)	7.1 Operaciones CRUD sobre resultados. 7.2 Ver máquinas conectadas. 7.3 Seleccionar tamaño de la muestra y forma de selección. 7.4 Validación Cruzada.

Tabla 3.2: Requerimientos de la capa de presentación.

3.2.2. Fuentes de Datos

La aplicación posee 3 repositorios de datos bien definidos, éstos se encuentran separados por el origen de los datos que contienen. Los registros médicos son la fuente de datos que contiene los registros electrocardiográficos, los registros de aplicación contienen los datos

propios de la aplicación web y finalmente los resultados de simulación contienen los resultados y configuración utilizada en cada una de las simulaciones realizadas utilizando el Framework.

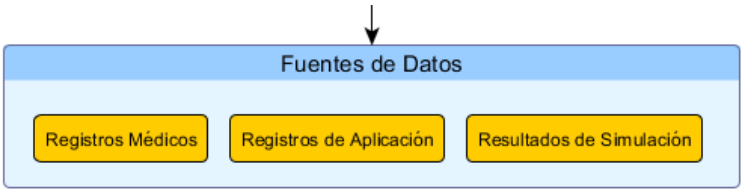

Figura 3.2: Arquitectura de Datos

Capa de Datos	Requerimiento
8 Registros Médicos	8.1 Proveer un modelo único de datos estandarizado. 8.2 Permitir múltiples fuentes de datos.
9 Registros de Aplicación	9.1 Modelo de datos de usuario. 9.2 Modelo de datos de notificaciones. 9.3 Operaciones CRUD sobre notificaciones y lectura sobre usuarios.
10 Resultados de Simulación	10.1 Modelo de datos de simulaciones. 10.2 Operaciones CRUD sobre las simulaciones.

Tabla 3.3: Requerimientos de la capa de datos.

3.2.3. Motor del Framework

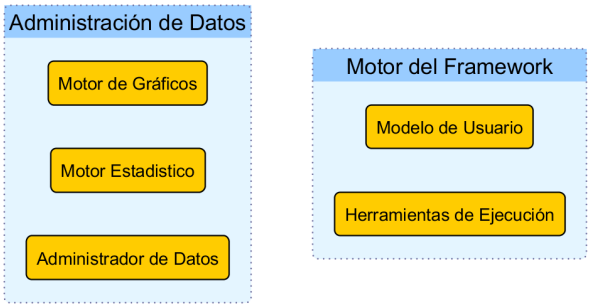

Figura 3.3: Requerimientos del Framework

El motor del framework es el componente más cercano al modelo de minería de datos de toda la arquitectura, este funciona mediante el patrón Inversipon de Control (IOC) que le permite al modelo abstraer gran parte de la lógica requerida para analizar un modelo, extracción acceso y guardado de datos. La abstracción proporcionada por el Framework es la principal motivación para su desarrollo y fuente de justificación.

Capa del Framework	Requerimiento
11 Motor de Gráficos	11.1 Modelo local de datos. 11.2 Controlador de gráficos.
12 Motor Estadístico	12.1 Modelo local de estadísticas. 12.2 Cálculo de estadísticas. 12.3 Controlador de estadísticas.
13 Administrador de Datos	13.1 Modelo local de datos. 13.2 Acceso a la API de la aplicación web. 13.3 Establecer conexión Socket con el administrador de datos web.
14 Herramientas de Ejecución	14.1 Administrar flujo interno del Framework. 14.2 Brindar base extensible para la creación del modelo de minería de datos.

Tabla 3.4: Requerimientos de la capa de datos.

3.2.4. Arquitectura Mínima Necesaria

Toda plataforma, servicio, sitio web, correo electrónico, etc. se requiere de tanto una arquitectura como un soporte de hardware que sostenga la ejecución del sofware de los servidores y servicios.

Figura 3.4: Flujo interno de la información del framework.

Basándose en el diseño de la Figura A.1 y lo detallado en las tablas 3.1, 3.2, 3.3 y 3.4. La arquitectura necesaria está compuesta de los elementos descritos a continuación

1. **Base de datos:** Servicio de almacenamiento y organización de datos, éste servidor debe manejar tanto los datos para las simulaciones, también de la inteligencia de negocio, como lo son datos de usuarios, perfiles, mensajes y otros elementos que la compongan.

2. **Servidor web:** Es el proveedor de servicios web, el encargado de responder y gestionar la interacción entre los usuarios de la plataforma.

3. **Servidor de aplicación:** Es el encargado de gestionar la comunicación entre el framework, la base de datos y el servidor web. Puede ser parte del servidor web, pero para mejorar el encapsulamiento, es mejor que se encuentre separado.

4. **Balanceador de carga:** En el caso que se quiera implementar un sistema de procesamiento en la nube, se requiere de un servidor que levante nuevas instancias. Para lo que se puede utilizar una de las diferentes plataformas de virtualización, lo que no es parte de esta discusión, sin embargo el concepto detrás de este servidor es la creación mantenimiento y destrucción de un ambiente en dónde se pueda ejecutar el framework y modelo del investigador.

5. **Framework:** Facilitador para la ejecución del modelo de minería de datos, provee de las herramientas básicas como la conexión al sistema de datos, almacenamiento de resultados, cálculo estadístico, entre otras funciones. Este puede ser ejecutado tanto en el

computador del investigador como en la nube, lo que permite ejecutar múltiples modelos al mismo tiempo y comparar fácilmente los resultados entre estos, su funcionamiento se puede apreciar en la Figura 3.4.

3.2.5. Flujo de las Comunicaciones de la Arquitectura

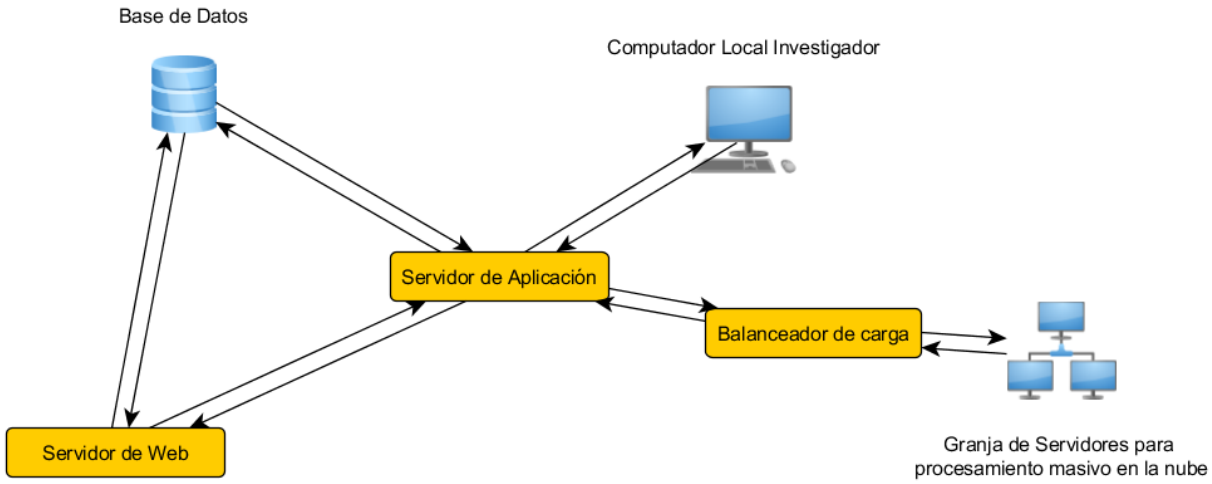

Figura 3.5: sads

Los componentes descritos anteriormente se comunican de forma bidireccional mediante el protocolo Transmission Control Protocol - Internet Protocol (TCP-IP). Esta comunicación es realizada por medio de un intercambio de paquetes estandarizados, el que puede ser realizado de forma sincrónica, envía el paquete y espera la respuesta, o de forma asincrónica, envía el paquete sin esperar la respuesta.

3.3. Discusión de Resultados y Propuestos

Gran parte del estudio de los requerimientos de la plataforma nace desde la investigación y levantamiento teórico de éstos, basándose en elementos destacados en la literatura. Sin embargo, no es hasta que la plataforma sea probada para la realización de una investigación que se podrá apreciar realmente la utilidad de la estructura diseñada y sus defectos. No es recomendable que la estructura de una plataforma en desarrollo sea rígida, en especial la de una plataforma que se encuentra en la fase de estudio de desarrollo como lo es la plataforma propuesta.

Es así como se entiende la falta del estudio de campo en éste capítulo al ser enfrentado desde la teoría y así es planteado en los objetivos de la memoria. Sin embargo, este enfoque posee riesgos y costos, como lo es el desarrollar un producto de baja calidad o que nadie desea, necesite.

Origen	Destino	Tipo	Sincrónico	Descripción
MP	BC, SA	Envío	No	Información de resultados parciales.
	BC, SA	Solicita	Si	Crear conexión vía protocolo socket y envía datos de autentificación.
SA	BD	Consulta	Si	Datos de registros médicos específicos.
	BD	Inserta	Si	Datos de resultados de simulación.
	MP	Envío	No	Inicialización de proceso de simulación.
	SW	Envío	No	Información sobre servidores locales conectados.
	SW	Envío	No	Información sobre simulaciones activas.
SW	BD	Consulta	Si	Información de usuario.
	BD	Consulta	Si	Información de registros y bases de datos almacenados.
	BD	Consulta	Si	Infromación y datos de resultados de simulaciones guardadas.
	SA	Envío	No	Información sobre nueva simulación.
	SA	Consulta	Si	Infromación sobre servidores locales connectados.
	SA	Consulta	Si	Información sobre simulaciones activas.

Tabla 3.5: Listado de los diferentes paquetes con los que los diferentes componentes de la aplicación se comunican entre si. **MP:** Máquina Procesadora, esta puede ser tanto el computador del investigador o una de las instancias levantadas por el balanceador de carga. **SA:** Servidor de Aplicación. **SW:** Servidor Web. **BD:** Base de datos. **BC:** Balanceador de Carga.

Por estos motivos es que se complementará este estudio teórico con los requerimientos prácticos, al desarrollar un modelo en el Capítulo 5 y se concluirá sobre su utilidad en la conclusión de ésta tesis.

Capítulo 4

Desarrollo de la Plataforma

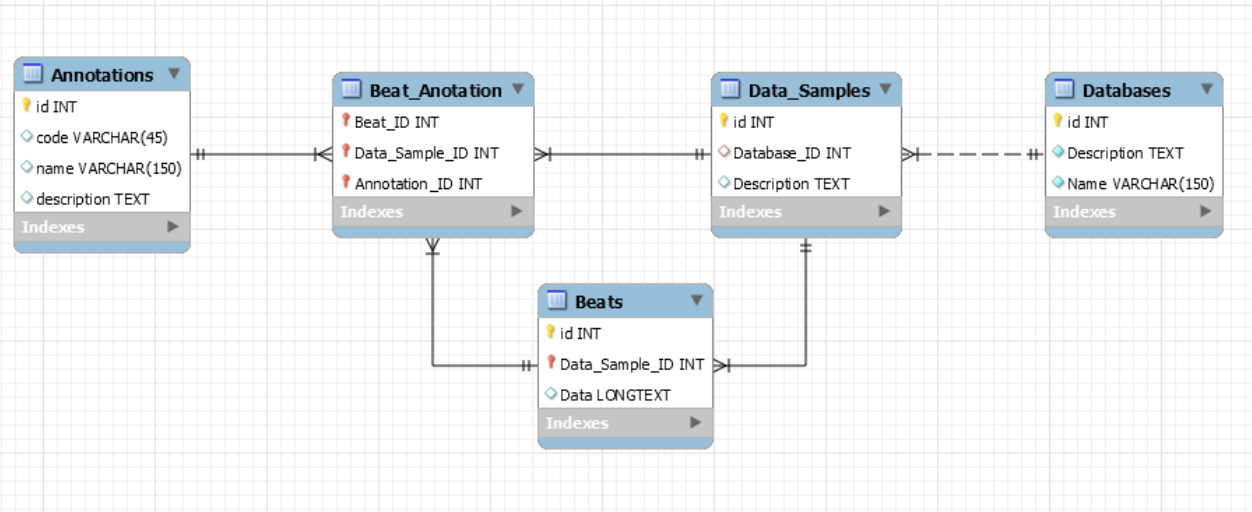

Figura 4.1: Diagrama de la estructura de almacenamiento de datos electrocardiográficos

4.1. Extracción de Datos

Los datos electrocardiográficos provienen de una base de datos pública, accesible a través del sitio web www.physionet.org, en donde se descargaron los registros de la base de datos de arritmias MIT-BIH. Estos datos se encuentran codificados bajo el estándar EDF+ [22] por lo que deben ser decodificados para ser utilizados con los motores comunes de procesamiento y/o ser leídos como archivos de texto simple.

La decodificación fue realizada utilizado las herramientas que physionet provee RDSAMP y RDANN, las cuales decodifican datos y anotaciones respectivamente.

Posterior a la decodificación los datos electrocardiográficos, éstos fueron cargados a la base de datos modelada para el sistema, dividiendo cada muestra médica por sus latidos, los que se relacionan con una anotación (clasificación) y una muestra. Cada muestra esta relacionada

con una colección de latidos y una base de datos a la que pertenece. Esta información puede ser vista en la estructura que la almacena en la siguiente sección.

4.2. Base de Datos

La base de datos de la plataforma esta compuesta por dos grandes grupos de datos, el primero se relaciona con la lógica de la aplicación, como lo son datos de usuarios, permisos, cuotas, entre otros. El segundo componente, almacena la información de registros médicos para la realización de los estudios. Ésta información es almacenada en cinco tablas que se pueden observar en la Figura 4.1.

Si bien la base de datos requiere más componentes de los que las figuras (Figura 4.1 y Figura 4.2) para que el sistema funcione en su totalidad, las tablas presentadas corresponden a las desarrolladas para la creación del prototipo. El que no requería la implementación de características como procesamiento distribuido, cuotas de utilización, almacenamiento de resultados persistentes, notificación, documentación, entre otros.

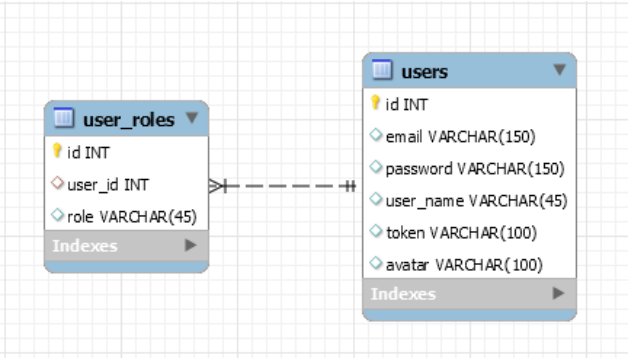

Figura 4.2: Diagrama de la estructura de almacenamiento de datos de usuarios

4.3. Plataforma

El prototipo de plataforma está compuesto por dos grandes servidores, un servidor web y una base de datos. El servidor web en el prototipo cumple las funciones de servidor web y servidor de aplicación, como se puede ver en la Figura 3.5. Adicionalmente como el prototipo no considera la implementación del procesamiento de modelos distribuido y en la nube, no se implemento la componente del balanceador de carga que se puede apreciar en dicha figura.

La plataforma se comunica con el computador del investigador mediante la ejecución de un programa que debe ser ejecutado antes de iniciar un test. Ésto por que la plataforma busca conexiones activas que estén disponibles para iniciar un test, si no existen conexiones a la plataforma retorna un error por falta de computadores para procesar la petición. Esta estructura permite, por ejemplo que se ejecuten múltiples pruebas o test en un mismo computador y la capacidad de procesar en forma remota.

El prototipo de la plataforma, fue programado en el lenguaje Java, utilizando el framework para aplicaciones web Spring y el administrador de dependencias Maven. Gracias a la utilización de dicho framework se logró implementar elementos de seguridad de nivel empresarial utilizando el componente Spring-Security. Éste es una extensión opcional del framework, creada y administrada por los programadores de Spring, que se encarga de la administración y validación de usuarios y formularios, de esta forma se evita reescribir componentes de seguridad y gracias a sus directivas de administración y configuración, se protege la aplicación contra ataques malintencionados conocidos.

4.4. Interfaz Web

La interfaz web es responsiva, cambiando la ubicación los diferentes componentes según el tamaño de la pantalla, la que posee múltiples vistas, que responden a las necesidades de la plataforma para interactuar con el usuario.

Las diferentes vistas requeridas para la un funcionamiento completo de la plataforma pueden ser clasificados en 3 grupos dependiendo del estado de desarrollo en el que se encuentran para término de ésta tesis. Éstos son 'implementados' vistas que se encuentran funcionales, tanto en su aspecto visual como en su funcionamiento y comunicación con el servidor web, 'mockup' maquetas de vistas, las que solo posee un valor visual de referencia, para posteriormente implementar el relleno con información dinámica perteneciente al usuario y finalmente las vistas propuestas, elementos que no fueron diseñados al no ser parte de los elementos críticos para el funcionamiento, pero que se consideran de suma importancia para el funcionamiento profesional de la plataforma.

1. Implementados

   (a) Log-In o Inicio de Sesión: Vista para la autentificación de usuarios (ver Figura C.1), en el que usuarios registrados en la plataforma ingresan su nombre de usuario y contraseña para acceder a ésta.

   (b) Dashboard o Panel de Control: Es la primera vista que los usuarios ven al ingresar a la plataforma, en dónde se visualizan macro datos de la ésta, como número de bases de datos, exámenes disponibles para hacer simulaciones y estadísticas de uso personales, como la cantidad de resultados disponibles, número de computadores conectados para el uso del investigador y cantidad de simulaciones que están corriendo (ver Figura C.2), ésta vista sirve como primera impresión y resumen del estado de la plataforma.

   (c) Formulario para Iniciar Test: Ésta vista es un formulario en el que él investigador configura elementos claves para crear un trabajo de procesamiento (ver Figura C.3) donde se envía para inicializar el trabajo de simulación con la configuración entregada.

2. Mockup o Maqueta

(a) Perfil de Usuario: Vista en dónde se puede visualizar datos personales del investigador, como tambipen estadísticas de sus trabajos (ver Figura C.4).

(b) Documentación: Vista de la documentación de la plataforma, elemento clave para el desarrollo de las investigaciones, tanto de referencia como fuente de aprendizaje sobre el uso de la plataforma (ver Figura C.6).

(c) Notificaciones: Vista en la que se incorpora un canal de comunicación entre la plataforma y el investigador. Así eventos como la finalización del procesamiento de una investigación, errores y notificaciones que pudiesen ocurrir en los diferentes actores que están involucrados en la generación de una investigación, tendrán un espacio en donde comunicar novedades en la plataforma (ver Figura C.5).

3. Propuestos

(a) Elementos críticos de negocio, como formas de pago, administración y recepción de números de tarjeta de crédito, facturación o en su defecto si la plataforma fuese orienta exclusivamente a la investigación sin fines de ser abierta al público en general, la asignación de cuotas de uso para cada investigador, entendiendo que los recursos son limitados y compartidos, por lo que se debe mantener control sobre ellos.

(b) Gráficos de presentación o vista de resultados, en esta vista de debe poder visualizar los elementos designados por el usuario para ser visualizados, los que suelen ser valores como por ejemplo la función de costo en una optimización.

(c) Resúmenes comparativos entre resultados, con variables programadas en forma genérica en el sistema, ésto por que los modelos suelen ir evolucionando con el pasar del tiempo, por lo que el valor por ejemplo de la función de costo para una misma simulación puede ser diferente si es que se decide cambiar los ponderadores de la función de costo. Lo que no es cierto en cuanto a la medición de la precisión y exactitud del modelo. De esta forma el nuevo modelo aún posee formas de ser comparado con los modelos previamente procesados, sin que el usuario tenga que preocuparse de éste problema desde el comienzo de la investigación.

4.5. Motor de Simulación

El motor de simulación es sencillo, como se mencionó anteriormente, éste requiere que el computador o servidor que va a procesar los datos se encuentre conectado al motor antes de que un nuevo trabajo de simulación sea desplegado. Éste comportamiento no es común en la industria, más bien puede ser considerado como un error de implementación, dicha afirmación es correcta, siendo la solución más común, la implementación de una cola de trabajo, en la que se apilan trabajos por realizar a la espera de recursos sean liberados o la llegada de nuevos recursos. Pero en el marco del desarrollo a nivel de prototipo de la plataforma, ésta no requería que se implementara una cola de trabajo, dado que solo se conectaría un computador al motor de simulación y el uso de ésta sería de baja demanda.

El motor de simulación recibe las peticiones de los computadores que quieren conectarse para recibir trabajos de procesamiento, mediante un intercambio de llaves, las que deben ser configuradas en el computador que se desea conectar al motor. De esta forma el motor puede autentificar que los trabajos que entregará al computador conectado son de propiedad del investigador que controla el computador, de la misma forma puede autentificar conexiones desde servidores de procesamiento en la nube, los que pueden ser creados bajo demanda y aceptar trabajos de cualquier investigador sin mermar en la privacidad y seguridad de la plataforma, al ser parte del control interno de la misma.

4.6. Implementación y Desarrollo del Prototipo

La implementación y desarrollo desde el prototipo a una solución profesional de la plataforma, posee grandes desafíos, uno de estos es su compleja arquitectura y las múltiples interacciones entre servidores y computadores que procesan las simulaciones, los que no necesariamente se encuentra en forma cercana. Por lo que se estima que un ingeniero recién egresado no tendría las herramientas requeridas para el desarrollo de la plataforma en el tiempo estimado. Ésto se profundizará en la Subsección 4.6.1.

Parte de los desafíos que posee la implementación de las características faltantes para completar la plataforma, es el framework con el que se desarrolló el prototipo de la misma. Esto considerando el caso que se decidiera continuar el desarrollo utilizando como base el prototipo programado para esta tesis.

El uso del framework Spring limita el espacio de recursos humanos que son capaces de continuar el desarrollo, debido a que para que un programador tenga un rendimiento óptimo de utilizando Spring, debe estar entrenado o capacitado en el uso de éste, lo que se puede ver materializado como un costo, tanto de entrenamiento en el caso que se decidiera capacitar al personal de desarrollo o en el salario extra que debiera recibir el personal ya capacitado, este costo no es considerado en éste estudio.

4.6.1. Costo del Desarrollo

Para la estimación de los costos del desarrollo de la plataforma, primero se describieron todos los elementos que le faltan al prototipo para completar la plataforma. Posteriormente, se estimó en base a una entrevista en profundidad con el gerente general (Jared Broad) de la firma prestadora de servicios de procesamiento de estrategias automáticas de inversión (QuantConnect) y junto con el gerente, se realizó una lectura de los registros de desarrollo que la compañía posee para estimar de mejor manera el tiempo requerido para el desarrollo de cada elemento.

El desarrollo del prototipo a aplicación funcional tiene un costo aproximado de 238 horas hombres (ver Tabla B.4). Estimación que fue determinada mediante un estudio del tiempo que tomó la implementación de tareas similares en el registro de desarrollo de la empresa QuantConnect.

Con los datos del tiempo estimado de desarrollo se puede estimar el costo del mismo. Según un estudio de sueldos patrocinado por el Instituto de Ingenieros de Chile [7], un Ingeniero civil en Computación con 5 años de experiencia, posee un promedio de sueldo de $1.666.666 pesos, por lo que en resumen la plataforma tendría un costo de $9.916.662 pesos, sólo en términos de salarios de los programadores.

A los valores anteriormente mencionados es necesario agregar costos adicionales, que varían según el escenario escogido para el desarrollo. Un ejemplo de esto es el arriendo de la oficina o espacio para el desarrollo. El que varía en sus requerimientos según la cantidad de programadores que trabajarán en forma simultánea en el proyecto. En el caso particular del Wic, éste posee instalaciones disponibles, que pueden ser habilitadas, sin costo, para el uso de los programadores necesarios.

4.6.2. Costo de Monitoreo y Mantenimiento

Como toda plataforma, ésta debe poseer un mantenimiento, es por este motivo que dentro del estudio de factibilidad de la plataforma se recomienda mantener al menos por medio tiempo (20 horas semanales) a un ingeniero capacitado en la plataforma, lo que tendría un costo adicional de $498.393 pesos, considerado el sueldo de un ingeniero recién egresado o con un año de experiencia [7].

Cabe destacar que el valor anteriormente mencionado, solo considera el costo de mano de obra para el mantenimiento de la plataforma, no así nuevos proyectos o características, con las que se puede mejorar la plataforma y los costos variables y fijos de operación que la plataforma posee.

Un servidor con las características necesarias para el almacenamiento de la base de datos posee un costo de USD 0.06/Hora [1], además el servidor web y el motor de procesamiento, poseen un costo de USD 0.03/Hora cada uno. A los precios anteriormente mencionados hay que agregar los costos variables de operación, que están ligados con la cantidad de horas y de servidores que son levantados para operar sobre demanda en las investigaciones que sean procesadas en la nube, el costo que posee un servidor que cumple con las características necesarias para el procesamiento es de USD 0.06/Hora. Considerando el precio del dólar a $704,08 pesos [2] la opereación de la platafroma posee un costo mensual de $39.653,79 pesos mensuales, más un costo variable de $42,24/Hora

4.7. Discusión de Resultados y Propuestos

Durante el proceso de desarrollo del prototipo del motor de la aplicación, surgieron ciertos pensamientos que posteriormente se fueron corroborando. Gran parte de la aplicación se concentra en el desarrollo de un sistema para el despliegue, almacenamiento, visualización de

[1] Precio por hora utilizando el proveedor www.digitalocean.com/pricing/ al 1 de Marzo del 2016

[2] Valor promedio observado por SII el mes de febrero 2016 www.sii.cl/pagina/valores/dolar/dolar2016.htm

los datos y resultados de las simulaciones. También existe una parte menor pero no menos significativa que está relacionada directamente con la interfaz web de la aplicación, como lo es la interacción con el usuario.

Mientras el desarrollo del prototipo avanzaba se comenzó a trabajar en un modelo que utilizaría el prototipo. Éste modelo se llevó a cabo utilizando el framework de Machine Learning y redes neuronales Tensorflow [2]. El framework provee múltiples herramientas dentro de las que se encuentra Tensorboard un visualizador de datos y del grafo de la red o modelo que se esta probando. La utilización de Tensorboard desplazó por completo la necesidad de poseer un motor estadístico, como también permite comparar múltiples modelos y ejecuciones de un mismo modelo.

Todas las herramientas antes descritas restaron importancia a la plataforma que en un principio tenía, siendo tan extremo el caso, que la utilización de la plataforma como iniciador del proceso de prueba de un modelo, comenzó a hacer que la ejecución misma del modelo tardara más. Esto se puede explicar por que la plataforma se comunica mediante el protocolo TCP-IP con el cliente, o computador que procesará el modelo. Dicho protocolo es notablemente más lento que leer los datos directamente desde el computador lo cuál en términos de rendimiento es de varios órdenes de magnitud más rápido.

Otro aspecto que se estimaba favorable para la plataforma, es el uso de una base de datos centralizada, lo que finalizó en términos de rendimiento siendo más lenta. La alternativa al uso de la base de datos es procesar los datos y guardarlos en el computador del investigador, ésto toma un tiempo de 30 minutos de programación más 20 minutos de procesamiento.

En conclusión, los múltiples beneficios que la plataforma presta puestos bajo el estudio de caso, parecen ser insuficientes para justificar el costo estimado que el desarrollo de ésta posee. Ésto nace tras la publicación y apertura al público en general del framework Tensorflow, el que es publicado en Noviembre 9 del 2015, tiempo para el que esta tesis, y el prototipo de plataforma ya se encontraban en desarrollo.

Capítulo 5

Caso de Uso, Modelo de Detección de Arritmias

Como análisis del modelo de detección de arritmia, se implemento un modelo de detección basado en DNNs, para el que se utilizó la interfaz TensorFlow [2] creada por investigadores de Google Research. Ésta puede ser ejecutada en una amplia variedad de sistema desde computadores de escritorio, teléfonos inteligentes a grandes instalaciones de servidores distribuidos.

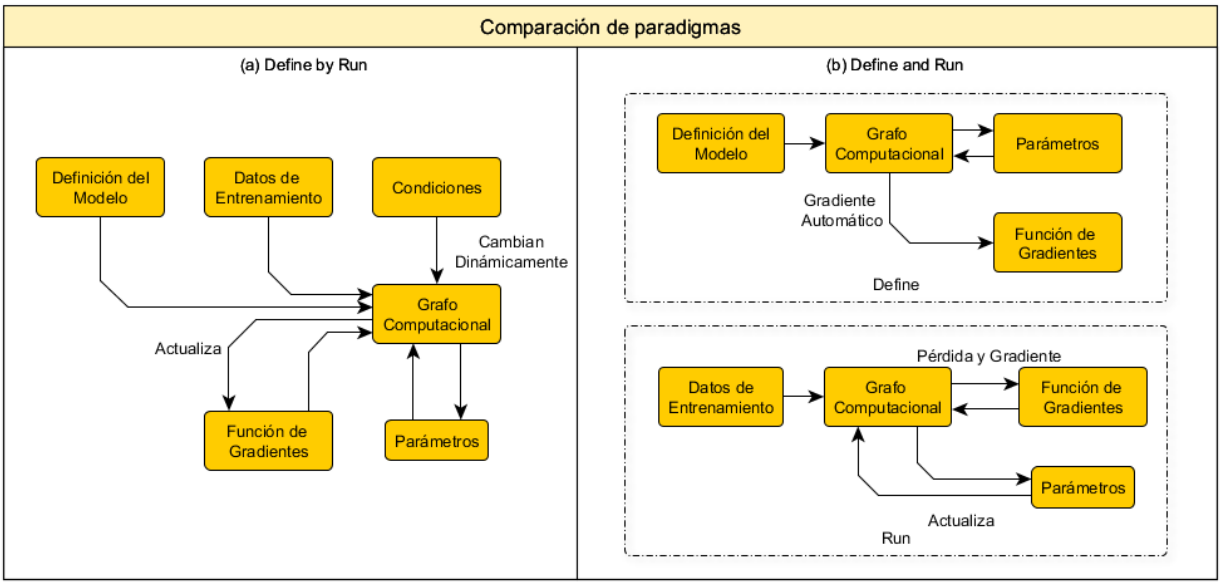

Figura 5.1: Comparación de paradigmas que implementan DNNs **(a):** Paradigma *'Define-by-Run'* **(b):** Paradigma *'Define-and-Run'* [35].

Actualmente existen otras interfaces o frameworks que resuelven el problema de la duplicación de código que las investigaciones que aplican DNNs tienen entre sí. Es así como estas herramientas pueden clasificarse dentro de dos categorías, *'Define-and-Run'* y *'Define-by-Run'*, Tensorflow pertenece a la primera categoría, y al ser la herramienta que actualmente Google utiliza para múltiples de sus plataformas, como reconocimiento de voz, búsqueda

en imágenes y búsqueda en texto, entre otras, es el principal motivo por el que se escogió esta interfaz para el desarrollo del modelo. Un ejemplo de "Define-by-Run.[es] Chainer [35], un framework escrito en Python que implementa el segundo paradigma mencionado, revisar Figura 5.1 con detalles de los macro procesos involucrados en cada uno.

5.1. Descripción del Modelo

La frecuencia cardíaca en reposo es comandada por el marcapasos natural del corazón, éste es el Nodulo Sinusal. Esta frecuencia tiene valores normales entre 60 y 100 pulsaciones por minuto [10], que en términos de frecuencia tendría un intervalo entre 1 y $\frac{5}{3}$ Hz. Con estos datos en mente se consideró suficiente una ventana de análisis de 5 segundos, ésta permite visualizar entre 5 y 8 pulsaciones cardíacas de forma simultánea.

Otra restricción que se es necesario tomar en cuenta al momento de realizar el modelo es la frecuencia de muestreo que se considerará para cargar el modelo. La frecuencia escogida es de 200 Hz, la que es inferior a los aproximados 358 Hz que la base de datos posee. Esta reducción en la frecuencia de muestreo le permite al modelo reducir una dimensión, al fijar el valor del tiempo para cara índice de datos.

De esta forma en vez de ser un conjunto de 1.748 tuplas de valores como se puede ver en la ecuación 5.1, el valor del tiempo se encuentra fijo en valores que incrementan de 5 en 5 ms, para lo que serían necesarias 1000 tuplas (ver ecuación 5.2). De esta forma para resolver el problema computacional se puede ingresar a la red neuronal un vector de largo 1000, como se puede ver en la ecuación 5.3.

$$[\{v_1, t_1\}, \{v_2, t_2\}, \{v_3, t_3\}, \{v_4, t_4\}, ..., \{v_{1748}, t_{1748}\}] \tag{5.1}$$

$$[\{v_1, 0\}, \{v_2, 5\}, \{v_3, 10\}, \{v_4, 15\}, ..., \{v_{1000}, 5000\}] \tag{5.2}$$

$$[v_1, v_2, v_3, v_4, ..., v_{1000}] \tag{5.3}$$

Se quiere estudiar la capacidad que las DNNs tienen para detectar y clasificar un latido en tiempo real. Para probar ésto, se utilizará una ventana flotante de tiempo, que irá progresivamente avanzando con pasos de 100 ms. Éste paso corresponde a un salto de 20 valores en el vector de datos propuesto en la ecuación 5.3.

Otra situación que el modelo debe enfrentar es que el valor del tiempo de las muestras obtenidas en la base de datos no coincide con el tiempo fijado para el vector en la ecuación 5.2. Éste problema es solucionado mediante una interpolación lineal entre los dos puntos más cercanos, un ejemplo de la interpolación utilizada es la ecuación 5.4 en la que v' corresponde al valor que debiera tomar en el vector simplificado para el tiempo t' con P_1 y P_2 los dos puntos muestrales más cercanos a P'.

Considerando los puntos : $P_1 = \{v_1, t_1\}$, $P' = \{v', t'\}$ y $P_2 = \{v_2, t_2\}$ con $t_1 < t < t_2$

$$v' = \frac{v_2 - v_1}{t_2 - t_1}(t' - t_1) + v_1 \tag{5.4}$$

La base de datos posee múltiples anotaciones, las que pueden o no estar relacionadas con la fisiología del latido, ver Tabla B.2 y Tabla B.1. De éstos dos casos se utiliza solo el primero, filtrando las anotaciones de la Tabla B.2 y clasificándose como latidos **anormales** todos los no están marcados como normales (N), las frecuencias de aparición de todas las anotaciones se encuentran listadas en Tabla 5.1.

Código	Frecuencia
N	**75.431**
A	**2.576**
V	**7.131**
/	**7.028**
f	**982**
Q	**31**
F	**803**
j	**229**

Código	Frecuencia
L	**8.075**
a	**150**
J	**83**
R	**7.259**
E	**106**
S	**2**
e	**16**
[	6

Código	Frecuencia
!	472
]	6
"	437
	612
|	128
x	193
+	1290

Tabla 5.1: Frecuencia de aparición de todas las anotaciones presentes en la base de datos de Arritmias MIT-BIH. En negrita se encuentran destacadas las anotaciones que corresponden a la fisiología del latido.

Con estos datos se decide armar una matriz de resultado de tres filas por mil columnas, lás dos primeras filas corresponde a índices o marcadores que indican que el latido que viene a continuación es del tipo sano (primera fila) o anormal (segunda fila). La tercera fila por otro lado corresponde a un marcador indefinido o punto interno del latido. Los dos primeros marcadores solo se activan en el caso que el punto representado sea un punto R del electrocardiograma (ver Figura 2.1). Esta configuración es una simplificación de la técnicas desarrolladas para la detección múltiple y clasificación de objetos en imágenes, un ejemplo de esto es el trabajo realizado por Dumitru Erhan [12], en dónde se propone como salida de la DNNs de múltiples cajas delimitadoras, con *'objetos detectados'* y sus puntuaciones de confianza, estos objetos posteriormente son clasificados utilizado una segunda red neuronal. Gracias a que la detección y clasificación del ECG es unidimensional, es decir, solo depende de la variable del tiempo, por lo que no se requiere encontrar los latidos completos para luego clasificarlos como en [12], pero si es necesario encontrar la frontera y la clasificación lo que justifica las tres dimensiones en las filas de la matriz de salida.

5.2. Desarrollo del Modelo

El modelo está inspirado en la capacidad mostrada para la comprensión y traducción de textos utilizando redes neuronales recurrentes [33, 6, 38, 4]. En caso particular de la técnica expuesta Kyunghyun Cho [6] en la que se describe cómo la utilización de dos redes neuronales recurrentes, pueden funcionar bajo un sistema de Encoder-Decoder, para transformar una serie de datos de largo indefinido, es decir que su largo no está fijo pero es finito, a un vector de transición que posee un largo fijo pero, luego éste vector sirve como entrada para la segunda red neuronal recurrente, la que entrega un vector de largo variable como salida, es posible ver un macro diagrama del grafo que genera este modelo en los apéndices Figura A.2.

La capacidad antes descrita es de gran utilidad para la realización de traducción de idioma natural, por que las frases pueden ser representadas por diferentes cantidades de palabras según el idioma el que son traducidos.

El modelo recibe como entrada una matriz con los valores de la series electrocardiográficas en el formato descrito en la ecuación 5.3 de las dos derivaciones que la base de datos provee. Posteriormente los datos son transformados para formar una matriz de 1.000 por 1.000 valores diagonal superior. Un ejemplo del como los datos que son reestructurados se puede apreciar en la transformación desde la Ecuación 5.5.

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 3 & 4 & 5 & 6 & 0 \\ 3 & 4 & 5 & 6 & 0 & 0 \\ 4 & 5 & 6 & 0 & 0 & 0 \\ 5 & 6 & 0 & 0 & 0 & 0 \\ 6 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{5.5}$$

Posterior a la creación de la matriz de 1.000 por 1.000 valores ejemplificada en la ecuación 5.5, se seleccionan las primeras 500 filas de la matriz, por lo que se obtiene una nueva matriz de 1.000 por 500 valores, ejemplificada en la ecuación 5.6.

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 3 & 4 & 5 & 6 & 0 \\ 3 & 4 & 5 & 6 & 0 & 0 \end{bmatrix} \tag{5.6}$$

Cada columna de la matriz anteriormente descrita alimenta en forma individual al modelo y dicho proceso se lleva a cabo una vez por cada derivación. Culminando el proceso de preprosesamiento de datos, al alimentar el modelo con una matriz de 2 por 500 datos.

$$\begin{gathered} Max_1 = Max(X_{(0,0)}, X_{(0,1)}) \\ Max_2 = Max(X_{(1,0)}, X_{(1,1)}) \\ Min_1 = Min(X_{(0,0)}, X_{(0,1)}) \\ Min_2 = Min(X_{(1,0)}, X_{(1,1)}) \\ FiltroTH = Relu(Exp(Max_1 * \alpha + Min_1 * \beta) + Bias_f) \\ FiltroDif = Relu(Exp((Max_1 - Max_2) * \alpha_{\mathrm{d}} + (Min_1 - Min_2) * \beta_{\mathrm{d}}) + Bias_{\mathrm{d}}) \\ th = Tanh(FiltroTh + FiltroDif) \end{gathered} \tag{5.7}$$

El modelo consta de dos grupos de redes neuronales que son entrenados de forma simultánea, el primer grupo posee como objetivo segmentar gran parte del espacio de datos que será ingresado al modelo, clasificando cada serie de vectores si corresponde o no un punto

R (ver Figura 2.1). Esta clasificación entrega una probabilidad (valor entre 0 y 1) la que es utilizada para disminuir la estimulación del segundo grupo de neuronas. Por lo tanto si el primer grupo de neuronas entrega el valor de 1, significa que el primer punto del vector ingresado es un punto R, de esta forma el segundo grupo de datos recibe los datos en forma intacta, sin embargo, si el primer grupo de neuronas indica que el primer punto no es un punto R, entregando una probabilidad de 0, todos los valores del vector de datos que recibirá el segundo grupo de neuronas serán 0, a esta probabilidad se le llamó 'threshold' o th en las ecuaciones, el que es calculado utilizando los dos primeros valores de cada derivación, los que son procesados en dos filtros para finalmente obtener el valor de th al aplicar la tangente hiperbólica sobre la suma de los filtros, para más detalles ver las ecuaciones 5.7.

$$[v_1, v_2, v_3, v_4, ..., v_{500}] \begin{bmatrix} th & 0 & \dots & 0 \\ 0 & th & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & th \end{bmatrix} \tag{5.8}$$

El comportamiento anteriormente descrito es ejecutado mediante la multiplicación de cada uno de los valores de los vectores de entrada del segundo grupo de neuronas por la probabilidad de que el punto inicial sea un punto R, esto es ejecutado mediante la multiplicación de cada vector por una matriz de 500 por 500 diagonal, en la que los valores de la diagonal corresponden al valor de th.

La idea principal detrás de éste diseño es controlar el estímulo que el segundo grupo neuronal recibe mediante el resultado que el primer grupo neuronal entrega. Es así como solamente se estimula el segundo grupo cuando la probabilidad de que el resultado de la primera red arroje que es altamente probable que el primer punto sea un punto R.

Una vez amplificados los datos, la matriz de 2 por 500 valores es reestructurada, condensando las dos filas en una sola, de esta forma el resultado es de un vector de 1 por 1.000 valores ejemplificado en la ecuación 5.9.

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix} \rightarrow [1, 2, 3, 4, 5, 6, 7, 8] \tag{5.9}$$

Como se fue mencionado en el Capítulo 2 los principales componentes que permiten estadísticamente realizar una clasificación de los latidos, es la distancia entre los puntos característicos de cada latido (P, Q, R, S y T) en la búsqueda por ayudar al modelo a aprender a utilizar estas distancias, se modeló una red que permitiera sacar provecho de ésto. El modelo consiste en tres filtros, compuestos por dos variables unidimensionales, llamadas sesgo s y ponderación p, éste filtro es posible apreciarlo en la ecuación 5.10.

$$Filtro = Tanh([v_1, v_2, v_3, v_4, ..., v_{1000}] + [s, \ldots, s]) \begin{bmatrix} p & 0 & \ldots & 0 \\ 0 & p & \ldots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \ldots & p \end{bmatrix} \tag{5.10}$$

$$RnnReady = Filtro_1 + Filtro_2 + Filtro_3 \tag{5.11}$$

Posteriormente el modelo lleva a cabo la suma del resultado de la aplicación en paralelo de los tres filtros, se espera que éste resultado sea capaz de entregar la información necesaria para clasificar el tipo de latido, dando como alternativa un latido normal o anormal.

El resultado de la operación anterior es ingresado a dos RNNs en forma paralela, compuestas por 1000 neuronas de entrada, 3 capas de profundidad. El tipo de neurona escogido para el modelo es la Long Short-Term Memory (LSTM) [19] las que se caracterizan por su capacidad para clasificar, procesar y predecir series de tiempo. De esta forma cada red aprenderá a identificar si el latido pertenece o no a su categoría, es importante mencionar que la primera red está encargada de identificar si es que el latido corresponde a un latido sano, por otro lado la segunda red es la encargada de identificar si es que el latido no es sano.

La idea detrás de ésta propuesta de modelo, es que si se requiere agregar más categorías, como la detección de diferentes tipos de enfermedades en el latido, se puedan incorporar más redes recurrentes en forma paralela.

$$Clasificador = RNN(RnnReady) * SoftMaxVar + SoftMaxBias \tag{5.12}$$

Posterior a la aplicación de las RNNs cada vector de resultado es multiplicado por una variable de tamaño 1 por 1.000 llamada $SoftMaxVar$ más una variable unidimensional o sesgo llamada $SoftMaxBias$ dando como resultado un vector unidimensional.

$$Softmax : P\left(y = j|x\right) = \frac{\mathrm{e}^{x^t w} j}{\sum_{k=1}^{K} \mathrm{e} x^t w_k} x \tag{5.13}$$

Los resultados de ambas RNNs son concatenados en un vector, al que se le aplica la función softmax, ecuación 5.13, el que permite transformar los resultados en probabilidades que se denominan $Logits$.

$$Logits = Softmax([Clasificador_1, Clasificador_2, 0]) \tag{5.14}$$

Finalmente, tras la aplicación del último filtro, se obtiene un valor del vector logístico, donde se vuelve a utilizar la variable th, la que es calculada por el primer grupo de redes neuronales, para añadir la información de la tercera columna del vector de resultado.

$$Resultado = (Logits + [0, 0, 1]) \begin{bmatrix} th & 0 & 0 \\ 0 & th & 0 \\ 0 & 0 & 1 - th \end{bmatrix} \tag{5.15}$$

5.3. Evaluación y Entrenamiento del Modelo

El modelo es evaluado mediante su ejecución en el framework Tensorflow, para el que se analizan variables claves, estas son la sensibilidad y especificidad tanto para la detección del punto R como para la clasificación del latido y la función de costo o objetivo que el modelo posee.

Como es mencionado anteriormente, desde la base de datos completa se elaboraron ventanas de intervalos móviles de 5 segundos, entregando un espacio de 864.088 archivos, de estos archivos se extrajeron 3 muestras, que representan el 40 %, 20 % y 40 %. Las dos primeras muestras representan el 60 % del espacio de la base de datos, este espacio es el utilizado para el entrenamiento y evaluación del modelo, de esta forma, el último 40 % del espacio muestral nunca fue utilizado para evaluar o estudiar el modelo, siendo utilizado solamente al final del proceso para una única evaluación final. La aplicación de ésta técnica de división de espacio posee como ventaja evitar el sobre ajuste del modelo y la realización de una evaluación en el marco de datos con los que el modelo no fue evaluado.

Muestras de Entrenamiento	345.636
Muestra de Evaluación	171.817
Muestras de Prueba	345.636
Total Muestras	864.088

Tabla 5.2: Tamaño de las muestras seleccionadas para cada fase del modelo.

$$Costo = ([1, 1, 1] - Y) * Resultado^T \tag{5.16}$$

Al modelo se le creó una función de costo, la que suma las probabilidades mal asignadas con respecto al resultado esperado Y, ver ecuación 5.16. El ajuste del modelo se llevó a cabo buscando minimizar el valor de la función de costo, el que fue realizado utilizando el algoritmo de optimización estocástica para modelos de aprendizaje Adagrad [11], debido a que durante las primeras iteraciones del modelo demostró converger en forma más rápidamente que los modelos que utilizaban un descenso por gradientes normal.

Como parte de la evaluación del modelo, se decidió utilizar como indicador de la evolución del modelo la suma de la función de costo evaluada en cada punto de la ventana de tiempo, ésto por la alta variabilidad que el indicador posee al observar los resultados de este en forma puntual.

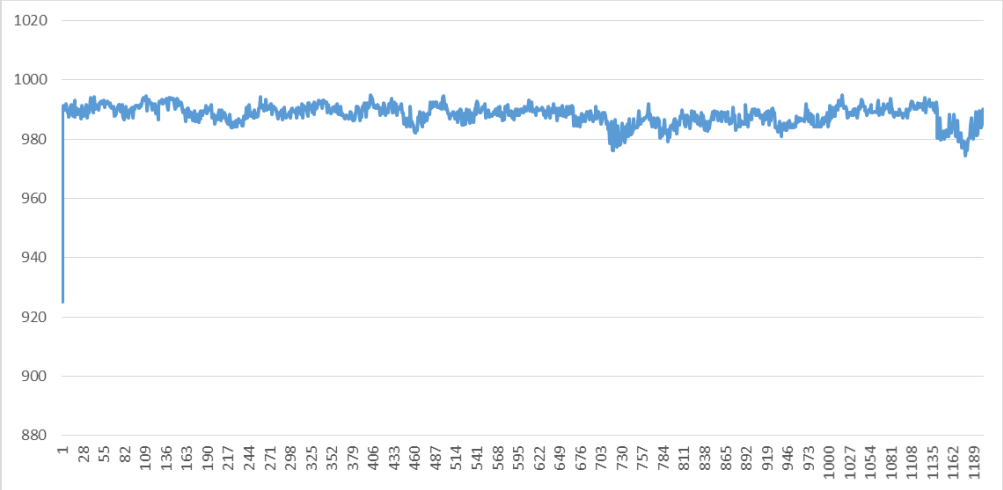

Figura 5.2: Gráfico del valor de la evolución del valor de la función de costo durante el entrenamiento de la red.

5.4. Discusión de Resultados y Propuestos

	Especificidad	Sensibilidad
Clasificación Latido	0,016129032	1
Detección punto R	0,003338898	0,999878733

Tabla 5.3: Resumen de los resultados obtenidos tras la ejecución final del modelo.

A pesar de los múltiples esfuerzos desarrollados durante el diseño del modelo, éste no mejoraba las capacidades de aprendizaje sin afectar considerablemente el tiempo necesario para procesar y entrenar una muestra de datos, esto es posible de observar en la Figura 5.2, en la que el modelo no posee un evidente ajuste en la suma de las funciones de costo por muestra procesada, lo que significa que el modelo no demostró grandes capacidades para mejorar la precisión de sus resultados.

Lo anteriormente descrito se vuelve aún más evidente al visualizar el comportamiento de la función de costo durante la fase de prueba del modelo Figura 5.3. Én dicho gráfico el valor del costo se encuentra evidentemente entorno al máximo posible, el que corresponde a 1.000, que sería el caso en que el modelo le asignara de forma erronea la probabilidad en todos los puntos de la muestra.

Los otros resultados que el modelo entrega son los de especificidad y sensibilidad, de estos resultados se pueden obtener varias conclusiones, primero la capacidad del modelo para reconocer un punto R es impresionante 99,98 %, sin embargo estos resultados contrastan con los que señalan la capacidad para indicar que un punto no es un punto R 0,33 %, resultado que puede ser explicado en parte por la gran diferencia en los tamaños de muestras que existe entre los puntos R y los que no lo son.

En cuanto al modelo de clasificación de latidos, una sensibilidad del 100 % demuestra un excelente rendimiento del modelo para aprender a identificar un latido sano, sin embargo, y

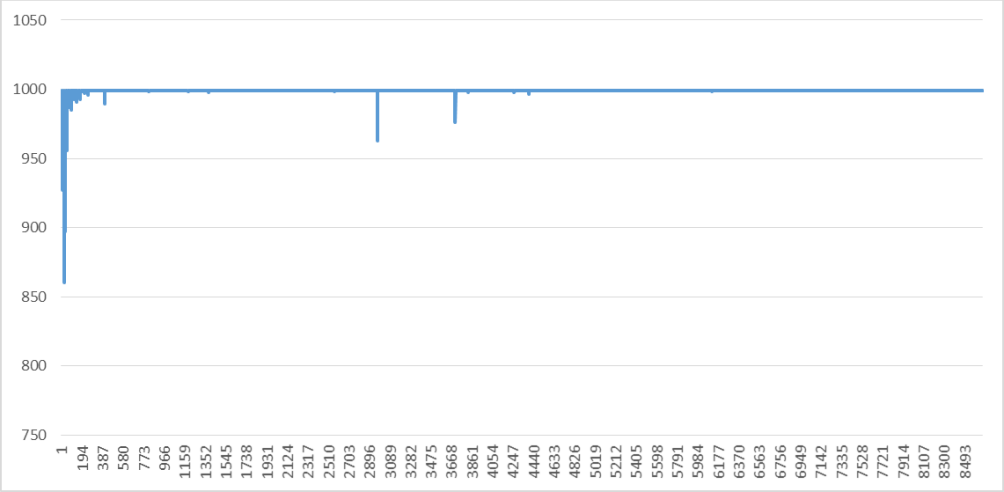

Figura 5.3: Resultados de la función de costo durante la fase de prueba del modelo.

al igual que con la detección del punto R, el modelo pareciera no ser preciso en la clasificación de un latido como anormal con una especificidad del 1.61 %.

Finalmente el modelo demostró tener la capacidad de ser preciso en cuanto a si un latido es marcado como sano, serlo realmente, sin embargo no se puede afirmar lo mismo cuando el latido es marcado como anormal, ya que existe una probabilidad no menor de que éste sea un latido normal clasificado en forma erronea. Esto se puede deber a que dentro de la categoría de latidos anormales se encuentran múltiples enfermedades, las que poseen características propias y para la profundidad configurada (3) puede ser difícil de aprender. De esta forma se deja propuesto evaluar y estudiar el comportamiento del modelo al agregar más RNNs, que se especialicen en la detección de una enfermedad en particular, ya que el modelo demostró ser capaz de ser preciso al afirmar que un latido era sano.

Conclusión

En el tiempo actual se ha declarado un movimiento o tendencia hacia el uso de redes neuronales en aplicaciones que antes era propias de algoritmos de modelos rígidos como regresiones o incluso se ha demostrado de gran utilidad la aplicación de redes neuronales en soluciones que los modelos tradicionales no pueden resolver como lo son aquellos problemas en los que las entras y salidas del modelo deben ser de largos diferentes, un ejemplo de esto es la traducción de frases en diferentes idiomas.

Es así como en ésta tesis se decidió estudiar los elementos necesarios para crear un framework que permitiera el estudio de modelos aplicados a la electrocardiografía, detectando anormalidades en el electrocardiograma.

5.5. Resultados Esperados

La hipótesis de investigación de esta tesis, planteaba la posibilidad de que tecnologías aplicadas a la detección y clasificación de objetos en imágenes, mediante el uso de redes neuronales, podía ser aplicada en forma satisfactoria a la detección de anormalidades en el electrocardiograma. Ésta hipótesis no pudo ser demostrada, ya que se obtuvieron bajos niveles de precisión en la detección de latidos anormales, no siendo capaz de superar modelos tradicionales y mixtos, que utilizan extractores de componentes principales para después entregar los resultados a un clasificador, el que puede ser una red neuronal o no. Ésto no implica que las técnicas no puedan ser aplicadas en forma satisfactoria en un futuro, más bien que los modelos planteados no son lo suficientemente complejos, en su modelo o entrenamiento, como para crear una red con la capacidad de clasificar un electrocardiograma con los datos en su forma básica.

Sin embargo, de ésta investigación se pueden desprender futuras líneas, como la propuesta en la siguiente sección.

Otro aspecto cubierto por los objetivos específicos de ésta tesis, son el levantamiento de los requerimientos para el desarrollo de un framework y el posterior desarrollo de un prototipo basado en los requerimientos levantados.

Dadas las similitudes del problema con la aplicación de la técnica de backtesting en finanzas, una entrevista en profundidad con el arquitecto y CEO de la empresa QuantConnect,

permitió ilustrar desde una posición fuera del problema de clasificación de arritmias, que la arquitectura para ambas plataformas era casi idéntica. Ambas recibían datos desde una fuente externa, que posteriormente deben ser distribuidos en una serie de nodos de procesamiento, que luego toman acciones y se deben realizar respaldos de estas acciones.

Con dicha entrevista en profundidad se realzó un listado de requerimientos mínimos, los que posteriormente sirvieron para la realización del estudio de factibilidad económica, como para la realización del prototipo de framework en forma exitosa.

Sin embargo, el estudio de factibilidad económica arrojó como resultado que el proyecto era inviable, para la etapa actual. Dado los costos de desarrollo y mantención que éste tendría. En particular el costo de desarrollo, se convierte en innecesario con la aparición del framework de código abierto de Google (Tensorflow), que cubre en gran medida los requerimientos mínimos de la plataforma y entregan un mejor rendimiento en términos de velocidad de procesamiento de datos, al utilizar una fuente local en vez de una centralizada.

El estudio realizado al utilizar el prototipo de framework con un modelo, mostró que la plataforma al ser usada en forma local no posee mucho valor y por el contrario, al transferir los datos a través de Internet, el proceso de la simulación avanzaba más lento de lo que podría avanzar si no se utilizara el framework, por lo que su valor solo aparece una vez que se implementan funcionalidades de procesamiento en la nube, las cuales serán discutidas en la siguiente sección.

5.6. Propuestas de Mejora y Aprendizajes para Futuras Investigaciones

De la experiencia adquirida luego de numerosos ensayos y errores en el desarrollo del modelo, se propone que en un comienzo de creen dos herramientas u objetos, el primero es un lector-administrador de datos, para si una vez obtenida la muestra de estudio y evaluación (sin utilizar la muestra de prueba o test), esto evitara que los diferentes modelos estudiados, presenten diferentes resultados, por que sus fuentes de datos son diferentes, se sugiere también realizar un objeto o herramienta, según el lenguaje en el que se esté modelando, que estudie en forma autónoma, las variables estadísticas clásicas como; sensibilidad, precisión, especificidad, curva ROC, entre otros. Para que posteriormente, ahorrar tiempo al no tener que volver a correr modelos por la aparición de la necesidad de tener un nuevo indicador, de esta forma se recomienda encarecidamente gastar tiempo al inicio de la investigación, en el desarrollo de un modelo autónomo y lo más definitivo posible en términos de cuales van a ser los indicadores claves y transversales a todos los modelos, para evitar tener que correr nuevamente los modelos, solo para poder comprar unos con otros.

Específicamente para tensorflow, es bueno utilizar una herramienta que fue puesta en práctica al final del período de investigación, la que permite guardar una sesión de entrenamiento. Esta herramienta guarda en forma persistente el estado de una red neuronal, por lo que se puede usar para dos fines, el primero es evaluar una entrada utilizando el estado guardado, o proseguir una sesión de entrenamiento, después de una falla. Lo que no es común, pero por

lo menos durante esta investigación, en el que tensorflow corría bajo un sistema virtualizado, ocurrió en varias ocasiones, en el que python se cerraba inesperadamente en medio de una sesión de entrenamiento. Las que por lo general se llevaban a cabo durante la noche, para posteriormente durante el día evaluar mejoras al modelo y volverlo a correr. Ésto significa que dentro del flujo de trabajo si la sesión de entrenamiento era interrumpida, al día siguiente no necesariamente se poseían resultados suficientemente concluyentes, debido a que estos tenían una alta probabilidad de estar sesgados por haber sido entrenados en set de datos pequeño.

Sesiones de entrenamiento, las redes neuronales requieren un alto grado de especialización, el que es difícil de lograr sin un gran set de datos y largas horas de entrenamiento, ésto es evidente por el tamaño del set de datos y por la cantidad de operaciones numéricas que deben ser ejecutadas durante el entrenamiento, es así como se privilegia el uso de operadores lineales, que permitan calcular fácilmente la matriz de gradientes de la red. Una analogía al nivel extra de complejidad que es agregado al pasar de optimización lineal a una optimización entera. Ésta analogía no esta lejos de parecerse al proceso que debe realizar la red durante la etapa de entrenamiento, dado que ésta debe optimizar una función objetivo.

Varios aspectos del entrenamiento pueden influir en los resultados obtenidos, es así como puede ser que el modelo de red neuronal que se propone, quede estancado en un mínimo local, sin poder seguir minimizando la función objetivo. Esto ocurre comúnmente cuando el paso o ratio de aprendizaje.[es] muy pequeño, no obstante si el paso o aprendizaje escogido es muy grande, al modelo le costara mucho fijar un aprendizaje, por que este puede ser sobrescrito por un 'outlier' fácilmente. Lo que en mi investigación entregó mejores resultados, es la utilización de una técnica conocida como 'decaimiento del aprendizaje', la que consiste en cambiar la capacidad de aprendizaje de la red en cada iteración, disminuyendo con el tiempo, es así como en su estado inicial, pasa rápidamente a un estado funcional, para después fijar el aprendizaje dado que los 'outliers' dejaran de afectar significativamente al modelo por la disminución del factor de aprendizaje. Para esto no hay formula ni valor mágico, solo ensayo y error, hasta que el modelo entregue los datos requeridos.

El modelo de decaimiento del aprendizaje utilizado es el siguiente:

$$RatioAprensizaje = 0{,}1 * (1 - Tanh(\frac{3 * \mathrm{i}}{Muestra})) \tag{5.17}$$

Siendo i la iteración o número de muestra con la que se está entrenando el modelo y el tamaño de la muestra, el total de muestras con las que el modelo será entrenado.

5.6.1. Mejoras en la metodología

Tal como se menciona anteriormente, la metodología con la que se trabajaba consistía en un ciclo diario de ensayo y error de hipótesis, que solo podían resolverse al día siguiente cuando se podía tener los resultados del entrenamiento con un set de datos de un tamaño considerable.

Esto implicaba que durante el día, el trabajo consistía en probar el funcionamiento de

pequeñas variaciones al modelo y su funcionamiento, por lo que durante la noche no siempre se estaba probando una sola hipótesis. Para entregar mayor precisión a una futura investigación se recomienda conseguir el financiamiento, para utilizar un servicio en la nube, que permita realizar el procesamiento simultáneo de múltiples hipótesis. Obteniéndose así una mayor presición a los efectos que cada iteración en el modelo tiene en la función objetivo.

Un ejemplo de esto sería arrendar un servidor en la nube de la empresa Digital Ocean, la que posee un costo de 0.03 dólares la hora, por lo que un estudio de en total 8 horas, con un tipo de cambio de 650 pesos por dolar, tendría un costo de 156 pesos. Este costo es marginal comparado con el costo de hora hombre de un investigador.

5.6.2. Nuevas líneas para la Investigación

Al final de la investigación de esta tesis, se ideó un modelo nuevo, que a diferencia de los modelos estudiados en el Capítulo 5, no simplifica los datos del electrocardiograma, si no que expande la dimensión del arreglo de datos de forma significativa.

Este procedimiento se lleva a cabo mediante la transformación de la serie de tiempo electrocardiográfica aplicando el filtro wavelet, el que convierte la serie de tiempo ingresada en un mapa de espectro, el que puede ser interpretado como un mapa de calor, en el que se muestra la serie de tiempo desglosada en una suma de ondas y sus amplitudes.

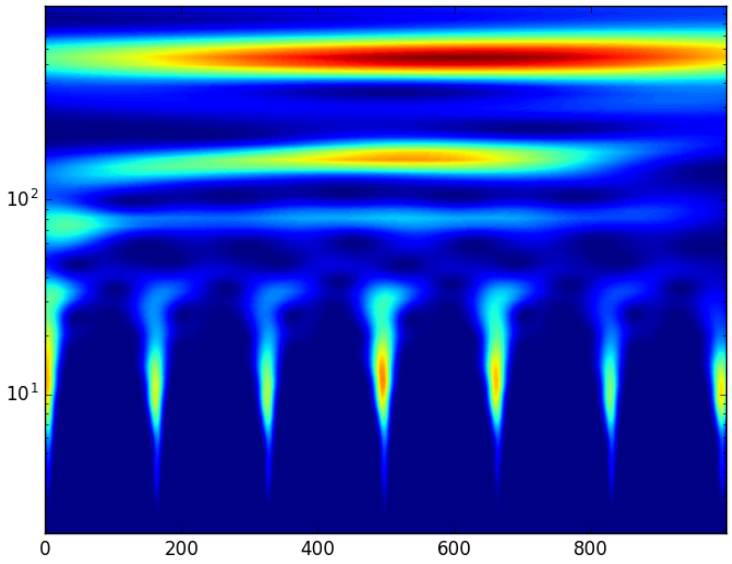

Figura 5.4: Frecuencia (hs) v-s tiempo (1/100 de segundo) de serie electrocardiográfica.

Luego de analizar el mapa de calor, es claro que hay que fijar dos niveles significativos, creando una banda de datos útiles para el problema, un ejemplo de una banda y aplicación del modelo a un solo latidos es el que es posible ver en la Figura 5.5.

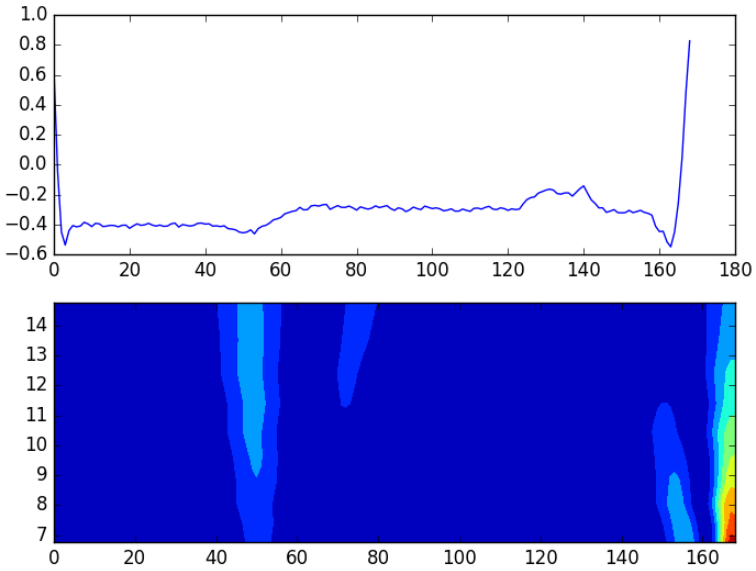

Figura 5.5: Arriba: Voltaje medido en mili volts versus tiempo medido en 1/100 de segundo, **Abajo**: Frecuencia medida en hz, versus tiempo medido en 1/100 de segundo.

Para finalizar, se recomienda estandarizar el largo de la matriz de entrada, acotando el problema a la resolución de una sola onda cardíaca (ver Página 50), sin embargo esta matriz de datos, que entrega la descomposición de wavelet, no es de un largo determinado, dado que ésta variará según el largo de la onda cardíaca. No obstante, este problema puede ser solucionado, considerando que la matriz de salida es una imagen, por lo que se podrían utilizar herramientas existentes para ajustar el tamaño de esta imagen a una dimensión fija de pixceles.

De esta forma, sería posible especializar cada zona de la red neuronal en una sola estructura del electrocardiograma, que es representada en el mapa de calor.

Glosario

Acoplamiento En informática, el nivel de interdependencia entre los elementos que componen el sistema, mientras mayor es el acoplamiento, menos independientes son los elementos. 23

API Interfaz de programación de aplicaciones o en inglés Application Programming Interface. 27

CNNs Redes Neuronales Convolucionales. 22

Correctitud Adaptación del la palabra inglesa 'Correctness', que hace referencia a si un algoritmo cumple con el objetivo para el que fue diseñado. 3

CRISP-DM Cross Industry Standar Process for Data Mining. xii, 6

CRUD De las siglas en inglés, Crear, Leer, Utilizar y Borrar. 25, 26

DB Base de Datos. 6

DBNNs Deep Belief Neural Networks. 21

Debug En formática los errores de programación son conocidos como 'bugs', por loq ue el proceso de eliminación de errores informáticos es descrito como 'debug'. 24

DNNs Redes Neuronales de Aprendizaje Profundo. xiii, 2, 10, 20–22, 37–39

ECG Electrocardiograma. ix, 1, 11, 12, 14, 15, 18, 24, 39

Electrodo Es un conductor eléctrico en contacto con una parte no metálica del circuito, en electrocardiografía el electrodo se encuentra en contacto con la piels plural. 14

Framework En desarrollo de software, es un conjunto de módulos que sirven como base para el desarrollo de aplicaciones. 4–10, 23–27

HI Health Informatics. 1

IOC Inversipon de Control. 26

LQT Desórdenes de repolarización. 15, 16

LSTM Long Short-Term Memory. 42

MINSAL Ministerio de Salud Chile. 2

NNs Redes Neuronales. 18–21

PSO Particle Swarm Optimization. 22

RBMs Restricted Boltzmann Machines. 21

RNNs Redes Neuronales Recurrentes. 20, 42, 45

ROC De las siglas en inglés, Característica Operativa del Receptor. 24, 25

Script Archivo de órdenes simples, de alto nivel que es usualmente interpretado por un programa o máquina virtual. 10

Socket En informática se refiere a la comunicación entre dos programas que pueden o no estar situados en el mismo computador. 27

TCP-IP Transmission Control Protocol - Internet Protocol. 28

Token Cadena de caracteres que tiene un significado para un sistema informático, ejemplo: autentificar un usuario. 25

UI User Interface. 25

WI Web Intelligence. 1

Bibliografía

[1] Physionet: The research resource for complex physiologic signals: Physiobank annotations.

[2] Martın Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous systems, 2015. *Software available from tensorflow. org*.

[3] Yaser S Abu-Mostafa, Malik Magdon-Ismail, and Hsuan-Tien Lin. *Learning from data*. AMLBook, 2012.

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[5] Mark L. Braunstein. *Practitioner's Guide to Heakth Informatics*. Springer, 2015.

[6] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[7] Colegio de Ingenieros de Chile A.G. Conexion Ingenieros. Estudio de sueldos de ingenieros 2015 y mercado laboral. 2015.

[8] Ministerio de Salud de Chile. Reporte de vigilancia de enfermedades no transmisibles (ent). Technical report, Ministerio de Salud de Chile, 2011.

[9] John Diaz, JF Olaya, and Consuelo Franky. Construcolectiva: Guía metodológica para la gestión de proyectos de software basados en metodologías ágiles, utilizando ambientes de desarrollo colaborativo. *Caso de estudio: GForge. Undergraduate final project, Pontificia Universidad Javeriana. Available in http://pegasus. javeriana. edu. co/˜ CIS0910IS05*, 2010.

[10] Dubin D Dubin. *Interpretación de ECG*. Cover Publishing Company, 2007.

[11] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.

[12] Dumitru Erhan, Christian Szegedy, Alexander Toshev, and Dragomir Anguelov. Scalable object detection using deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2147–2154, 2014.

[13] Adam Gacek and Witold Pedrycz. *ECG signal processing, classification and interpretation: a comprehensive framework of computational intelligence*. Springer Science & Business Media, 2011.

[14] İnan Güler and Elif Derya Übeylı. Ecg beat classifier designed by combined neural network model. *Pattern recognition*, 38(2):199–208, 2005.

[15] Rajarshi Gupta, Madhuchhanda Mitra, and Jitendranath Bera. *ECG acquisition and automated remote processing*. Springer, 2014.

[16] John E Hall. *Guyton and Hall textbook of medical physiology*. Elsevier Health Sciences, 2010.

[17] J. Heaton. *Artificial Intelligence for Humans, Volume 3: Deep Learning and Neural Networks*. Artificial Intelligence for Humans. CreateSpace Independent Publishing Platform, 2015.

[18] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, 29(6):82–97, 2012.

[19] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[20] International Telecommunication Union (ITU), Scientific the United Nations Educational, and Cultural Organization (UNESCO). The broadband commission for digital development. Technical report, UNESCO, 2015.

[21] S Karpagachelvi, M Arthanari, and M Sivakumar. Ecg feature extraction techniques-a survey approach. *arXiv preprint arXiv:1005.0957*, 2010.

[22] Bob Kemp and Jesus Olivan. European data format plus edf+, an edf alike standard format for the exchange of physiological data. *Clinical Neurophysiology*, 114(9):1755–1761, 2003.

[23] Richard Klabunde. *Cardiovascular physiology concepts*. Lippincott Williams & Wilkins, 2011.

[24] Catherine Klersy, Annalisa De Silvestri, Gabriella Gabutti, François Regoli, and Angelo Auricchio. A meta-analysis of remote monitoring of heart failure patients. *Journal of the American College of Cardiology*, 54(18):1683–1694, 2009.

[25] Bert-Uwe Köhler, Carsten Hennig, and Reinhold Orglmeister. The principles of software qrs detection. *Engineering in Medicine and Biology Magazine, IEEE*, 21(1):42–57, 2002.

[26] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[27] SZ Mahmoodabadi, A Ahmadian, and MD Abolhasani. Ecg feature extraction using daubechies wavelets. In *Proceedings of the fifth IASTED International conference on Visualization, Imaging and Image Processing*, pages 343–348, 2005.

[28] George B Moody and Roger G Mark. The impact of the mit-bih arrhythmia database. *Engineering in Medicine and Biology Magazine, IEEE*, 20(3):45–50, 2001.

[29] Philips. St segment and arrhythmia monitoring. 9 2015.

[30] PWC. "best"practices: The elusive benefit. 2 2014.

[31] Danny Shapiro. 8 reasons your next car might be your first personal supercomputer. Enero 2016.

[32] Danny Shapiro. Eyes on theroad: How autonomous cars understand what they'r seeing. Enero 2016.

[33] KK Shukla and Arvind K Tiwari. *Efficient Algorithms for Discrete Wavelet Transform: With Applications to Denoising and Fuzzy Inference Systems*. Springer Science & Business Media, 2013.

[34] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[35] Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton. Chainer: a next-generation open source framework for deep learning.

[36] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1653–1660. IEEE, 2014.

[37] Rüdiger Wirth and Jochen Hipp. Crisp-dm: Towards a standard process model for data mining. In *Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining*, pages 29–39. Citeseer, 2000.

[38] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.

Apéndice A

Ilustraciones y Gráficos

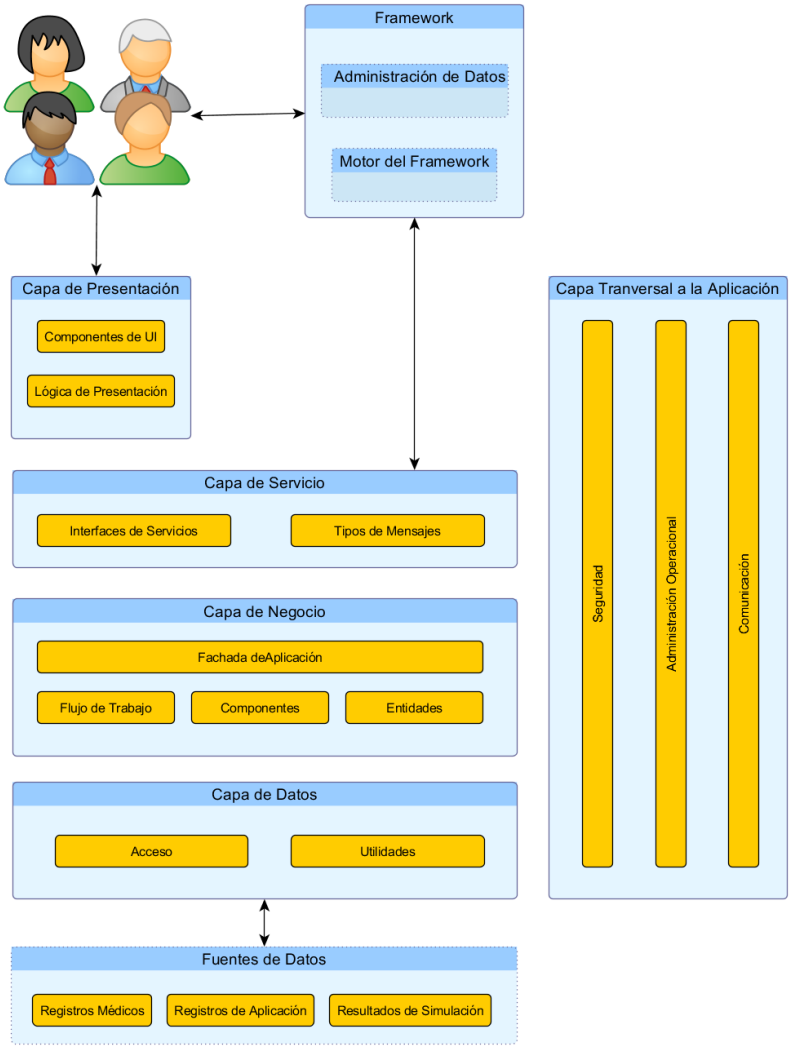

Figura A.1: Arquitectura web y del framework integradas.

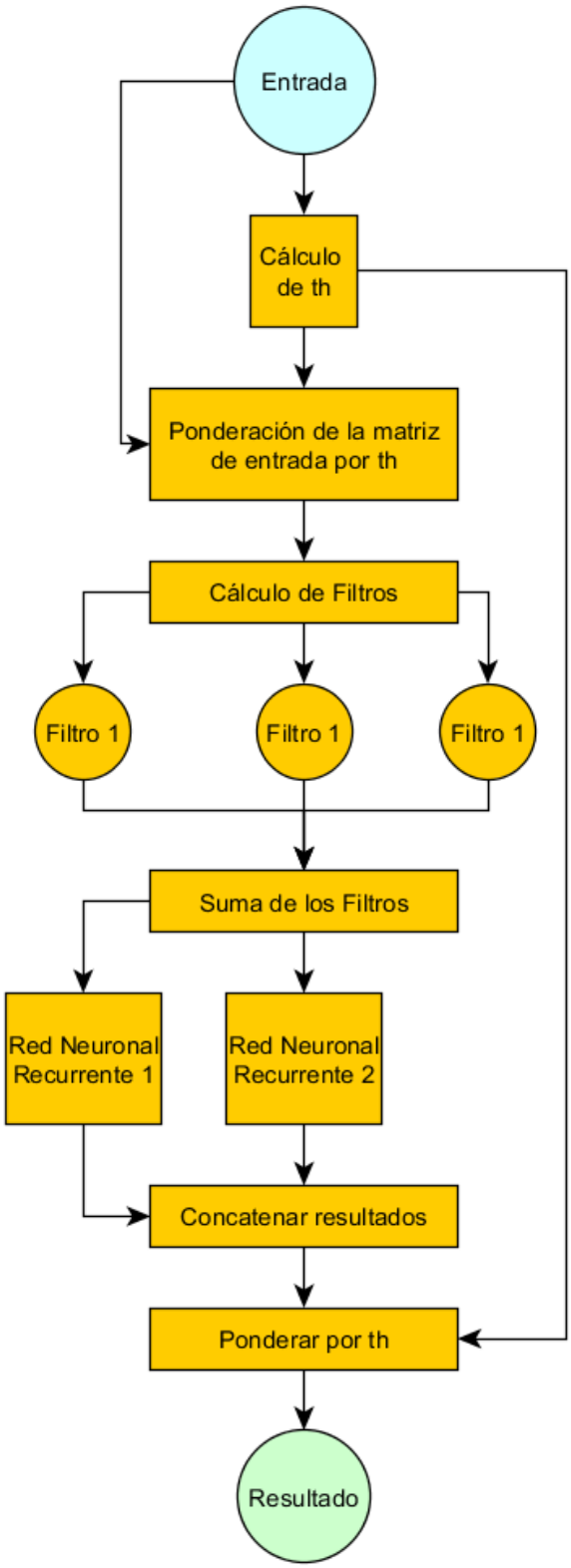

Figura A.2: Grafo del modelo de detección de arritmia implementado.

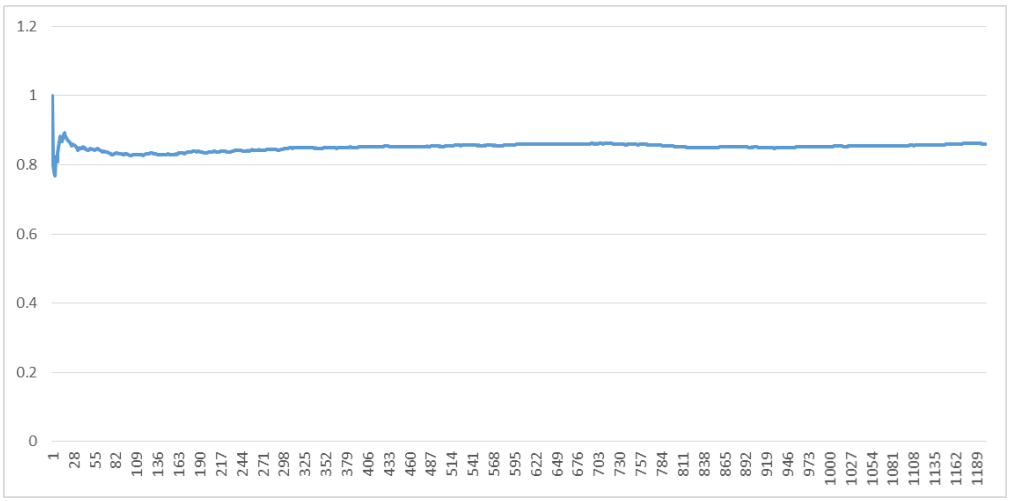

Figura A.3: Evolución del valor de la sensibilidad en el modelo durante el entrenamiento.

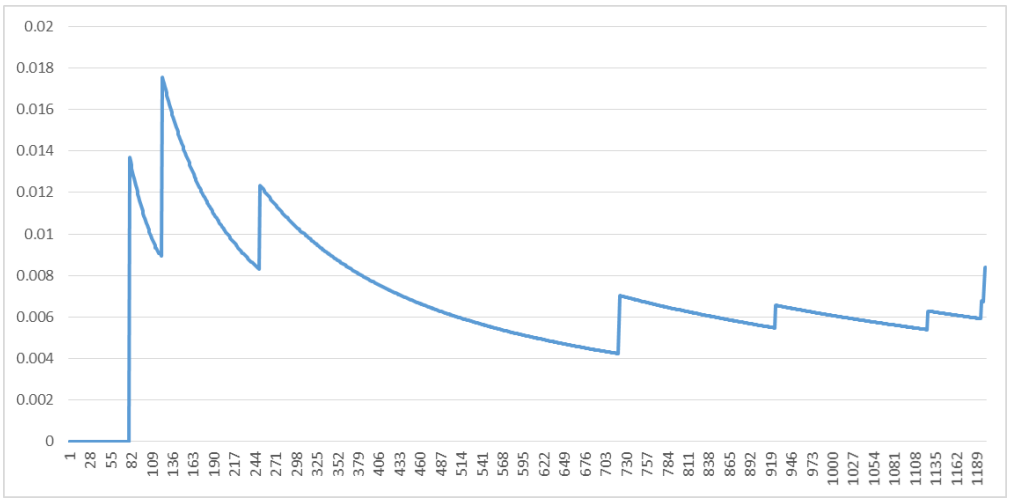

Figura A.4: Evolución del valor de la especificidad en el modelo durante el entrenamiento.

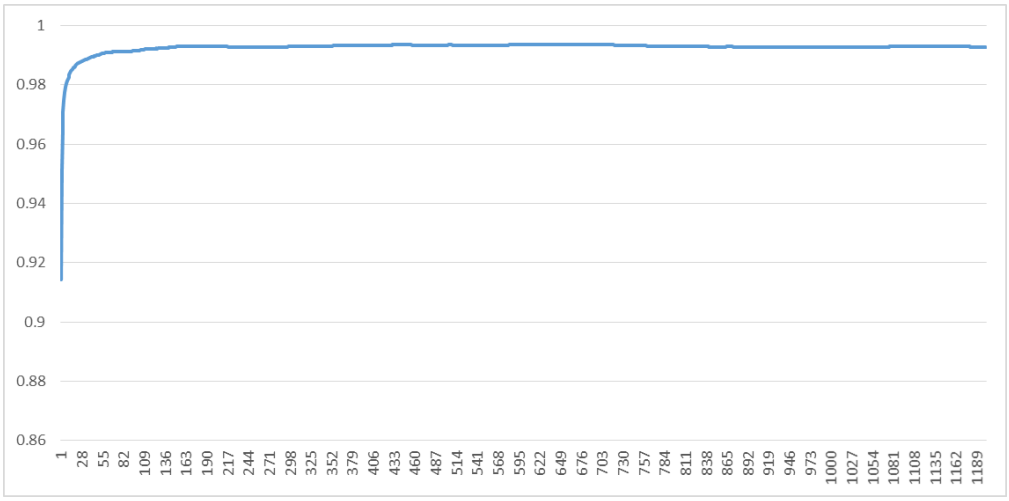

Figura A.5: Evolución del valor se la sencibilidad para *th* durante el entrenamiento.

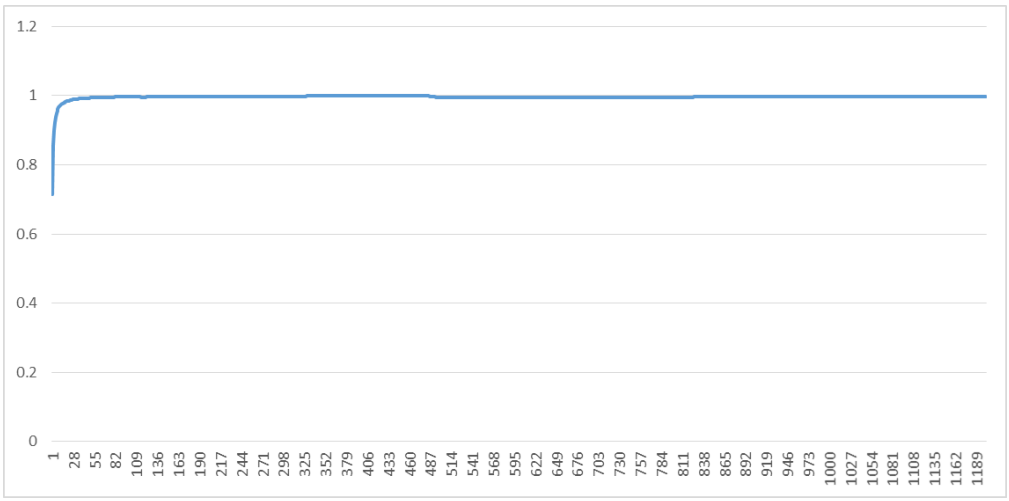

Figura A.6: Evolución del valor de la especificidad de *th* durante el entrenamiento.

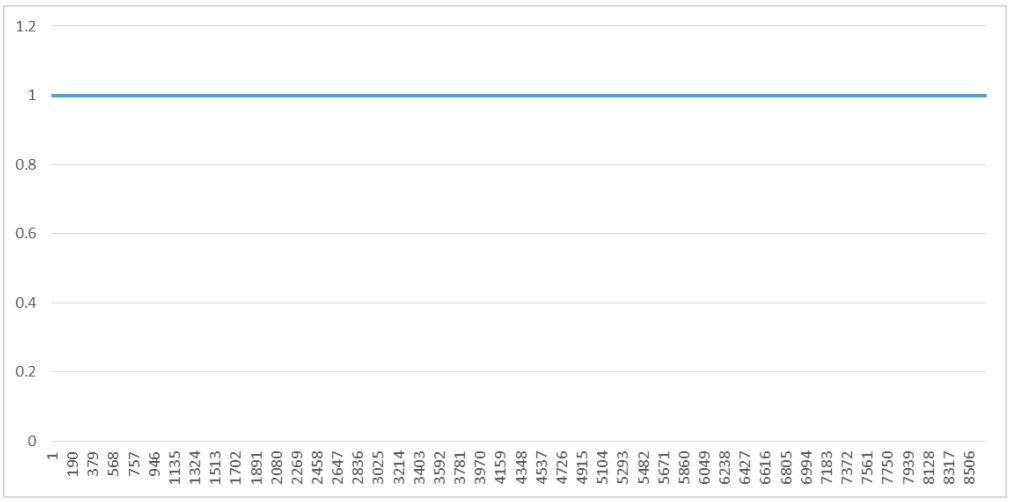

Figura A.7: Evolución de la sensibilidad que el modelo entrega en la fase de prueba.

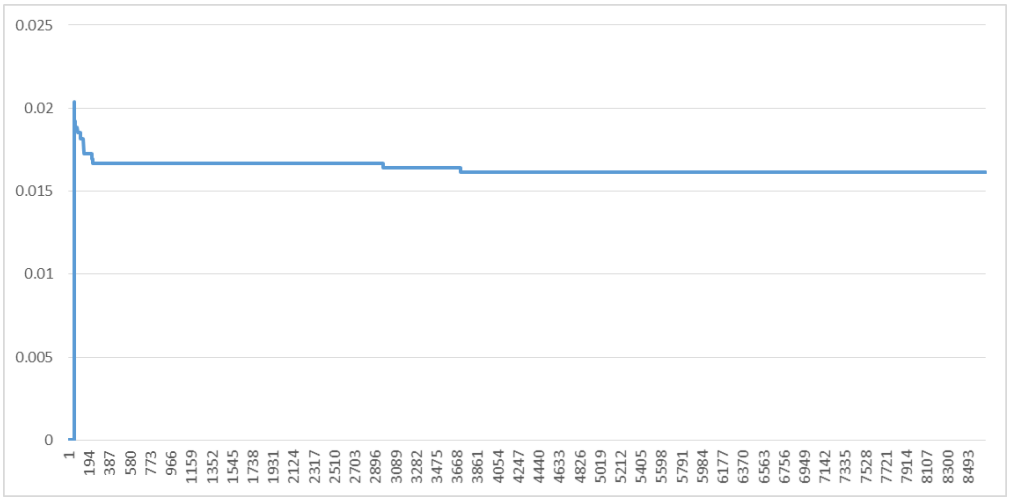

Figura A.8: Evolución de la especificidad que el modelo entrega en la fase de prueba.

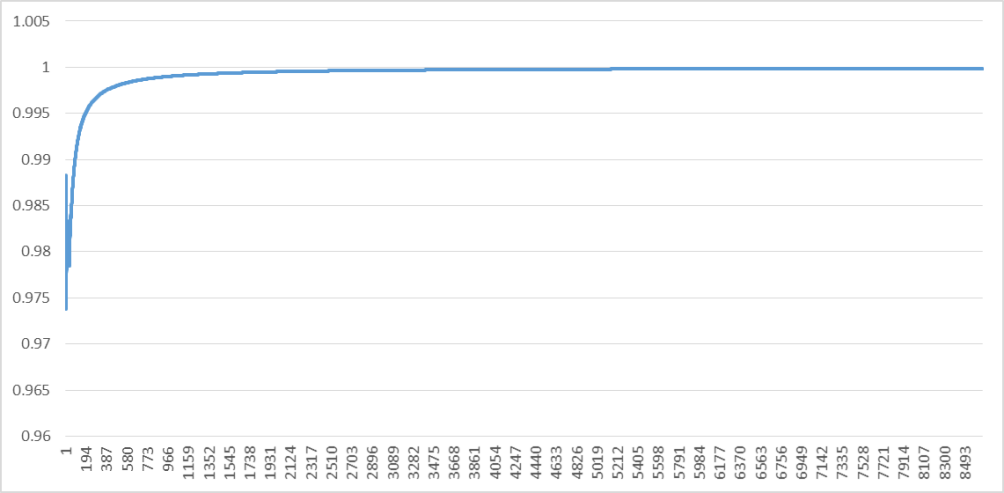

Figura A.9: Evolución de la sensibilidad de *th* en la fase de prueba.

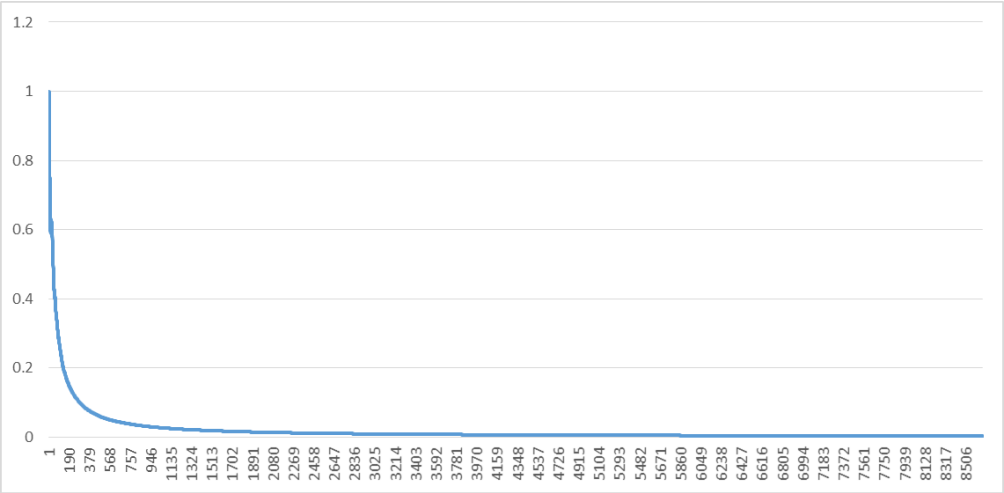

Figura A.10: Evolución de la especificidad de *th* en la fase de prueba.

Apéndice B

Tablas

Código	Descripción
N	Normal beat (displayed as · by the PhysioBank ATM, LightWAVE, pschart, and psfd)
L	Left bundle branch block beat
R	Right bundle branch block beat
B	Bundle branch block beat (unspecified)
A	Atrial premature beat
a	Aberrated atrial premature beat
J	Nodal (junctional) premature beat
S	Supraventricular premature or ectopic beat (atrial or nodal)
V	Premature ventricular contraction
r	R-on-T premature ventricular contraction
F	Fusion of ventricular and normal beat
e	Atrial escape beat
j	Nodal (junctional) escape beat
n	Supraventricular escape beat (atrial or nodal)
E	Ventricular escape beat
/	Paced beat
f	Fusion of paced and normal beat
Q	Unclassifiable beat
?	Beat not classified during learning

Tabla B.1: Anotaciones relacionadas con la fisiología del latido, fuente [1]

Código	Descripción
[	Start of ventricular flutter/fibrillation
!	Ventricular flutter wave
]	End of ventricular flutter/fibrillation
x	Non-conducted P-wave (blocked APC)
(	Waveform onset
)	Waveform end
p	Peak of P-wave
t	Peak of T-wave
u	Peak of U-wave
‘	PQ junction
’	J-point
^	(Non-captured) pacemaker artifact
|	Isolated QRS-like artifact
~	Change in signal quality
+	Rhythm change
s	ST segment change
T	T-wave change
*	Systole
D	Diastole
=	Measurement annotation
@	Link to external data

Tabla B.2: Anotaciones no relacionadas con la fisiología del latido [1]

	Agrupamiento no Supervisado	Regresión	Clasificación	Predicción	Robótica	Visión	Optimización
Mapa Autoorganizado	★★★				★	★	
Prealimentación		★★★	★★★	★★	★★	★★	
Hopfield			★			★	★
Máquina de Boltzmann			★				★★
Red de Creencia Profunda			★★★		★★	★★	
Prealimentación Profunda		★★★	★★★	★★	★★★	★★	
NEAT		★★	★★		★★		
CPPN					★★★	★★	
HyperNEAT		★★	★★		★★★	★★	
Redes Convolucionales		★	★★★		★★★	★★★	
Red de Elman		★★	★★	★★★			
Red de Jordan		★★	★★	★★	★★		
Redes Recurrentes		★★	★★	★★★	★★	★	

Tabla B.3: Tipos de redes y su Dominio [17]

Item	Costo HH
Consulta a la base de datos para almacenar resultados de procesamientos	3
Consulta a la base de datos para notificaiones	2
Consulta a la base de datos para perfil del investigador	2
Consulta a la base de datos sobre resultados de procesamientos	8
Consultas a la base de datos sobre cuotas de usuario	2
Escribir documentación del sistema	80
Estructura de base de datos para almacenar información de notificaciones	4
Estructura de base de datos para almacenar información de perfil del investigador	2
Estructura de base de datos para la administración y control de cuotas de usuario	3
Implementación de cola de trabajo para motor de procesamiento	8
Implementación de estadisticos locales generalizados	8
Implementación de sistema de muestreo para gráficos	4
Implementación de sistema de recepción de resultados	8
Implementación de un balanceadro de carga	8
Implementación de un sistema procesameinto estadistico local	8
Implementacion en vista de documentación del sistema	24
Implementacion en vista de gráficos de resultado	24
Implementación en vista de historial de resultados	4
Implementacion en vista de notificaciones	2
Implementación en vista de perfil del investigador	2
Implementación en vista de sistema de asministración y control de cuotas de usuario	8
Implementación de un sistema de administración de recursos bajo demanda	24

Tabla B.4: Costos en horas hombre del desarrollo de los elementos restantes del prototipo.

Apéndice C

Capturas de Pantalla

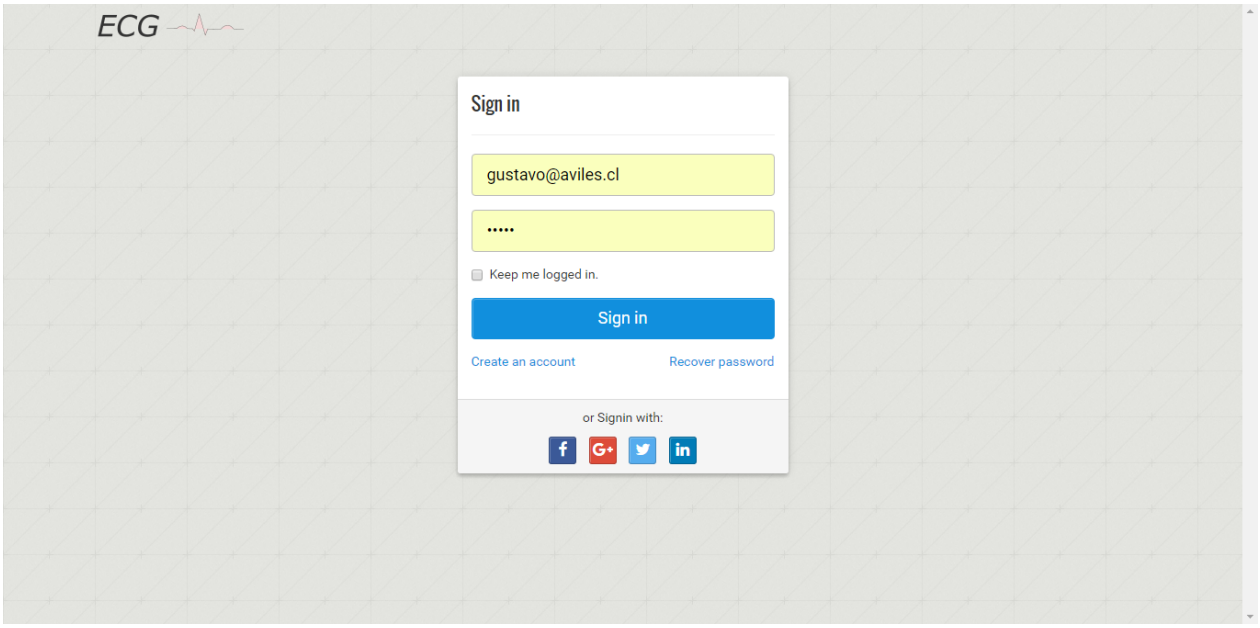

Figura C.1: Captura formulario de entrada a la platafroma.

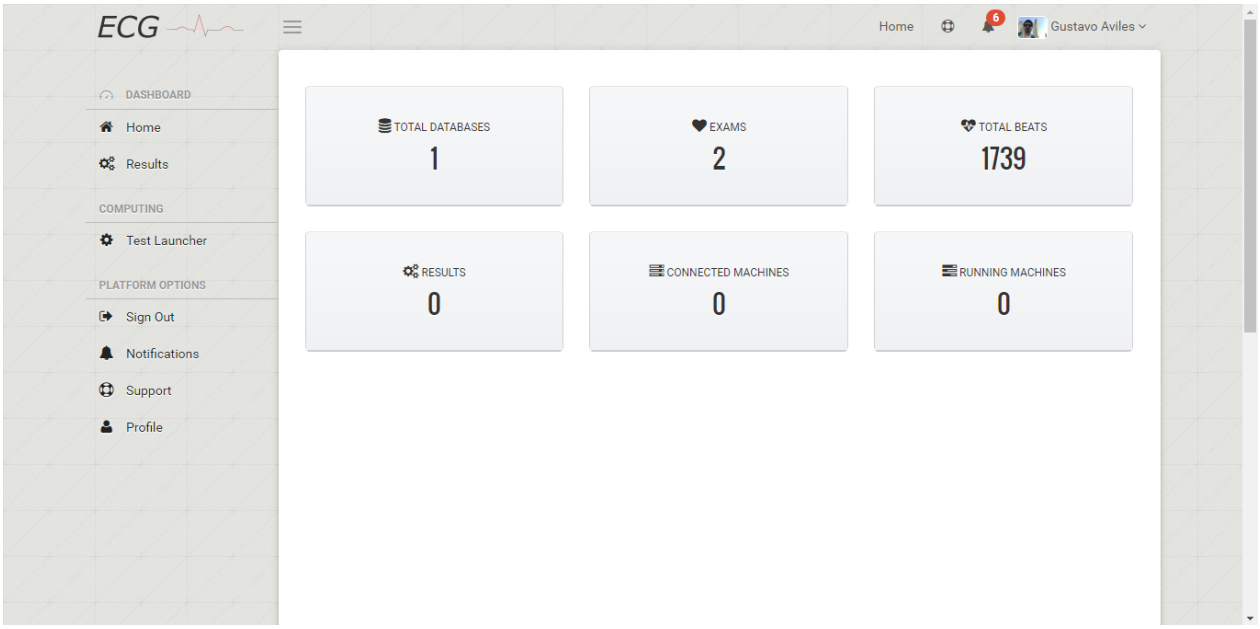

Figura C.2: Captura vista principal de la plataforma.

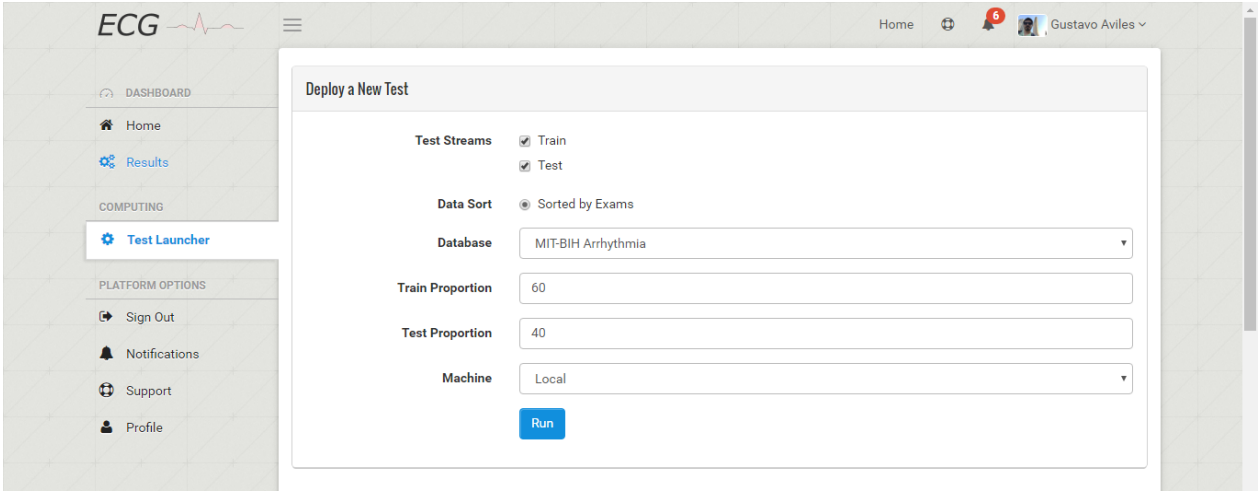

Figura C.3: Captura formulario lanzamiento de test.

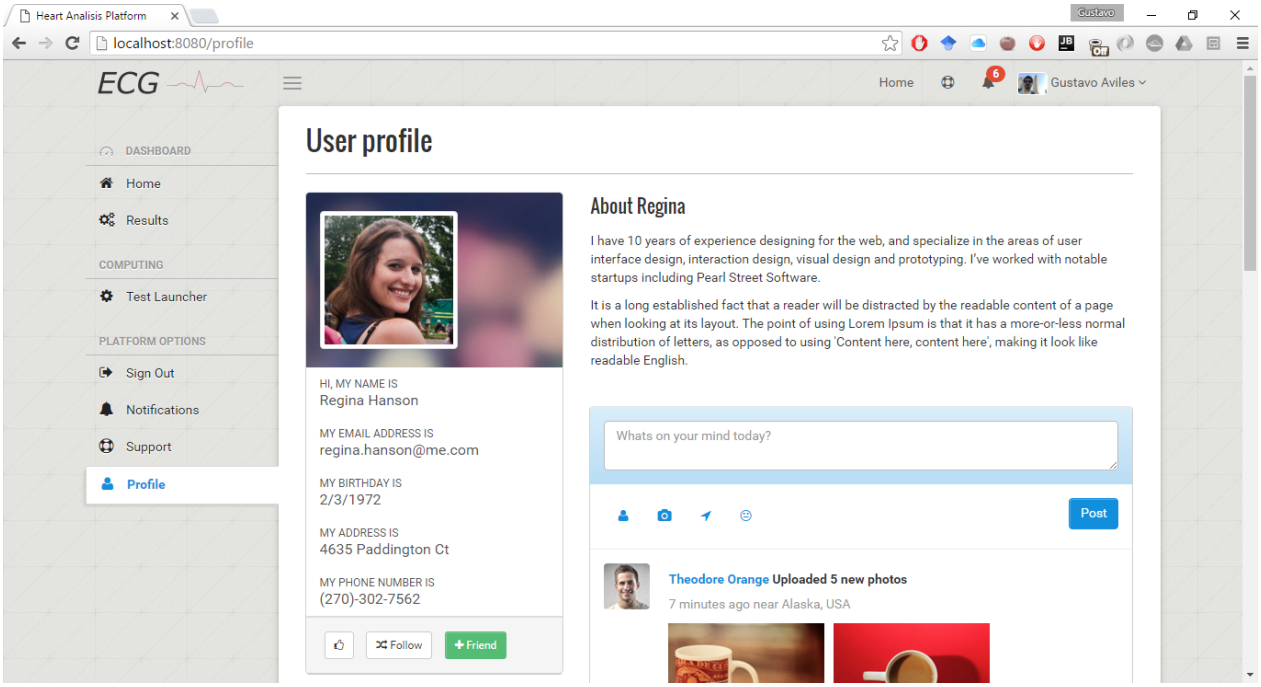

Figura C.4: Captura modelo de perfil de usuario.

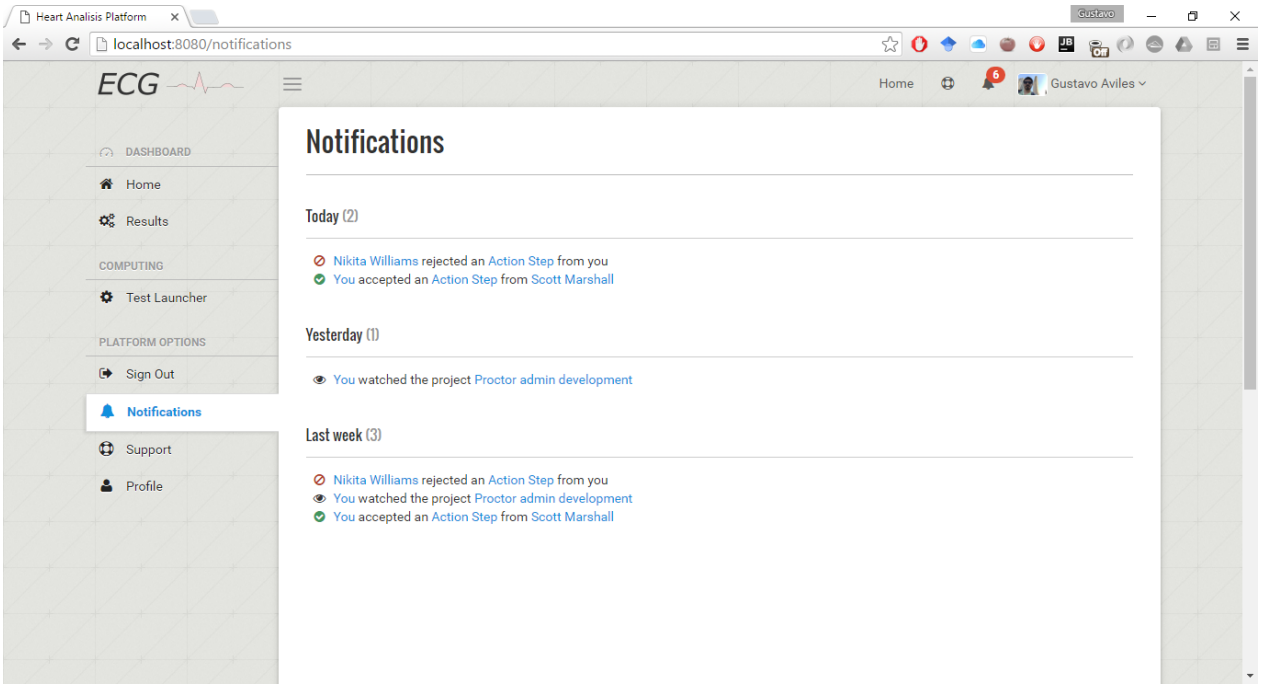

Figura C.5: Captura modelo de notificaciones.

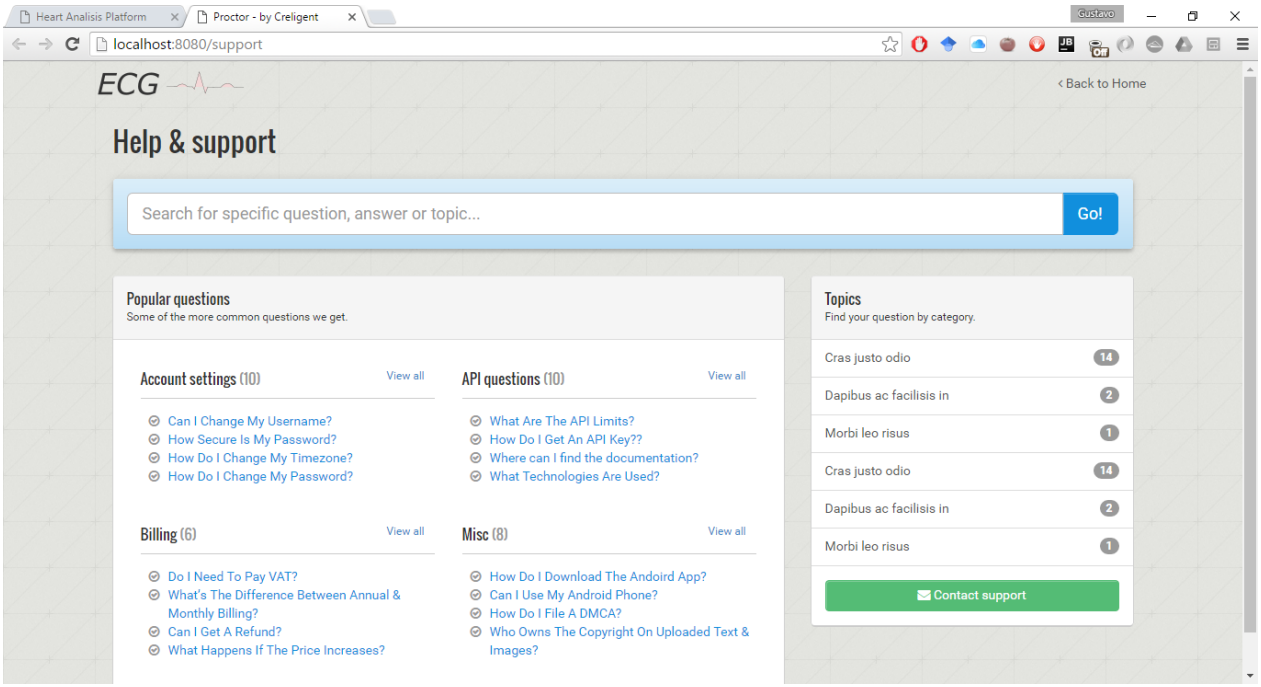

Figura C.6: Captura modelo de documentación.

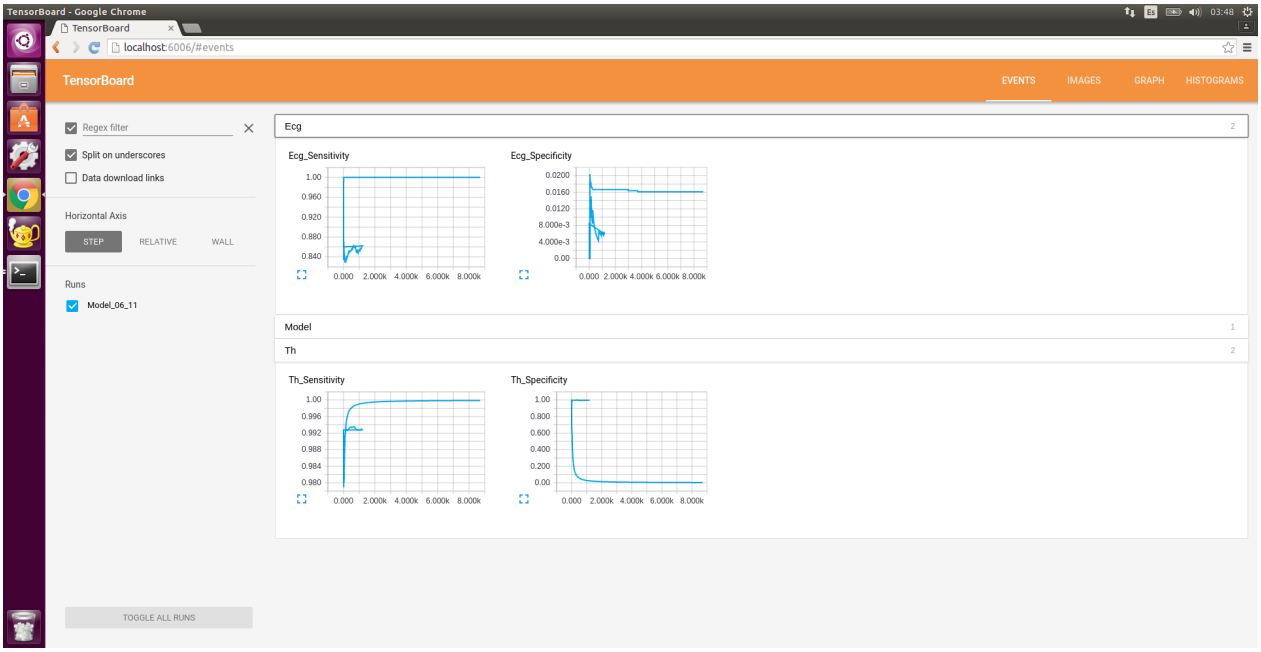

Figura C.7: Captura de pantalla de la interfaz gráfica Tensorboard, mostrando la vista de eventos.

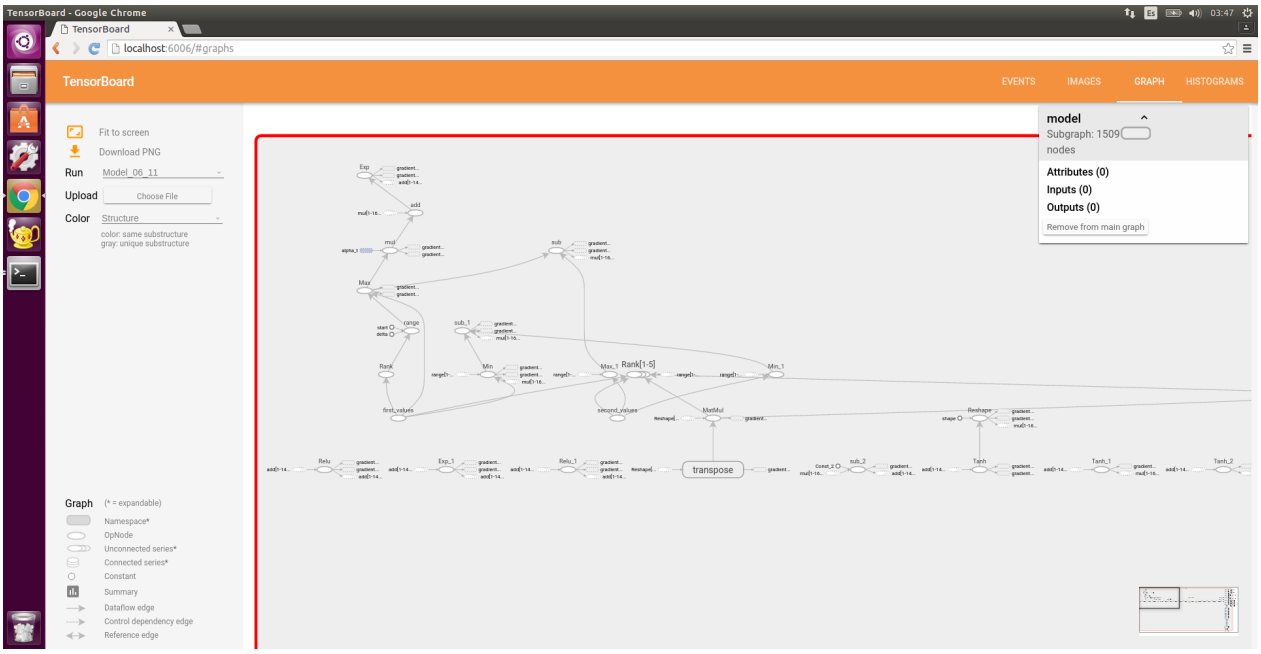

Figura C.8: Captura de pantalla de la interfaz gráfica Tensorboard, mostrando la vista parcial del grafo desarrollado en ésta tesis.