

UNIVERSIDAD CENTRAL DE VENEZUELA

FACULTAD DE CIENCIAS

ESCUELA DE COMPUTACIÓN



**DISEÑO E IMPLEMENTACIÓN DE UNA APLICACIÓN WEB DE
MANEJO DE REGISTROS Y ESTADÍSTICAS DE
ELECTROCARDIOGRAFÍA**

Trabajo Especial de Grado presentado por
Alvaro Marquina

ante la Facultad de Ciencias de la
Ilustre Universidad Central de Venezuela
como requisito parcial para optar al título
de: **Licenciado en Computación**

Con la tutoría de: Prof. Esteban Álvarez
Prof. Robinson Ribas

Mayo-2017

Caracas-Venezuela

Escuela de Computación

UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN



**DISEÑO E IMPLEMENTACIÓN DE UNA APLICACIÓN WEB DE
MANEJO DE REGISTROS Y ESTADÍSTICAS DE
ELECTROCARDIOGRAFÍA**

Trabajo Especial de Grado presentado por
Alvaro Marquina
ante la Facultad de Ciencias de la
Ilustre Universidad Central de Venezuela
como requisito parcial para optar al título
de: **Licenciado en Computación**

Con la tutoría de: Prof. Esteban Álvarez
Prof. Robinson Ribas

Mayo-2017
Caracas-Venezuela

UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACION



ACTA DE VEREDICTO

Quienes suscriben, Miembros del Jurado designados por el Consejo de Escuela de Computación, para examinar el Trabajo Especial de Grado, presentado por el Bachiller Alvaro Marquina, portador de la cédula de identidad 17.263.959, con el título **“Diseño e implementación de una aplicación web de manejo de registros y estadísticas de electrocardiografía”**, a los fines de cumplir el requisito legal para optar por el título de Licenciado en Computación, dejan constancia de lo siguiente:

Leído como fue dicho trabajo por cada uno de los Miembros del Jurado, se fijó el día 3 de Julio del 2017, a las 9 AM, para que su autor lo defienda en forma pública, en las instalaciones de la Escuela de Computación, mediante la exposición oral de su contenido, y luego de la cual respondieron satisfactoriamente a las preguntas que le fueron formuladas por el Jurado, todo ello conforme a lo dispuesto en la Ley de Universidades y demás normativas vigentes de la Universidad Central de Venezuela. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobarlo.

En fe de lo cual se levanta la presente Acta, en Caracas a los tres días del mes de julio del año dos mil diecisiete, dejándose también constancia de que actuaron como Coordinadores del Jurado los Profesores Tutores Robinson Rivas y Esteban Álvarez.

M.R.

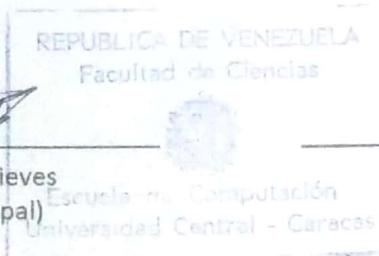
Prof. Robinson Rivas
(Tutor)

E. Álvarez

Prof. Esteban Álvarez
(Tutor)

A. Nieves

Profa. Adelis Nieves
(Jurado Principal)



A. Morales

Profa. Ana Morales
(Jurado Principal)

Agradecimientos

A mis padres, Malta Rivero y Noel Marquina.

A mi gran amiga Valentina, por su gran apoyo y ayuda en este proyecto

A mi novia Jessica Rengifo, por su apoyo y motivación.

A mi tutor, el Dr. Esteban Álvarez, por la dedicación al proyecto y su impecable guía y paciencia.

A los médicos especialistas, Ana Elisa y Federico Moleiro (†) por su asesoría y conocimientos a lo largo del proyecto.

A los empleados de la Sección de Cardiología del Instituto de Medicina Tropical de la UCV, por toda la colaboración brindada.

A mi gran amigo Rubén Rondón, y su prodigiosa mente.

Al gran Gordom Shumway por su constante inspiración durante la ejecución del presente proyecto.

A mi universidad, facultad y escuela, por todos los conocimientos y experiencia adquirida.

A mis amigos, que de un modo u otro contribuyeron con mi crecimiento personal y profesional.

A cada uno de los profesores que contribuyeron a mi formación profesional.

RESUMEN

El funcionamiento del corazón humano se puede medir a través de un electrocardiograma, en él podemos monitorizar los latidos y su frecuencia, su estudio se hace a través de la serie temporal de intervalos RR, que no es más que la secuencia cronológica de los intervalos temporales entre latidos o complejos cardíacos consecutivos (distancia en milisegundos entre ondas R); gracias al estudio de su variabilidad (VFC, o variabilidad de la frecuencia cardíaca) se puede conocer el estado general de salud del corazón del paciente; en el presente proyecto se ofrece una herramienta bajo software libre para apoyar en esta labor al especialista o cardiólogo, gracias a que nos permite acceder de manera ágil y amena a los registros de los pacientes, su historial, además, con la implementación de herramientas para un estudio analítico y gráfico, facilitando la generación de un reporte que soporte el diagnóstico dado por el medico especialista. Esta herramienta fue realizada para acceder de manera remota, gracias a su implementación web; para ello se utilizaron herramientas del lado del servidor con el lenguaje Python y el framework Django, y del lado del cliente o navegador utilizando HTML, CSS, Javascript y Bootstrap. El resultado de esta implementación es una aplicación sencilla de usar pero a la vez poderosa.

Palabras clave: corazón, electrocardiograma, electrofisiológicas, aplicación web, python, django, javascript, bootstrap, VFC, frecuencia cardíaca, variabilidad de la frecuencia cardíaca, estadísticas, indicadores temporales, indicadores frecuenciales, web.

Índice general

Lista de figuras	9
Lista de tablas	11
Introducción	13
1. Planteamiento del problema	15
1.1. Planteamiento del Problema	15
1.2. Objetivo General	17
1.3. Objetivos Específicos	17
1.4. Alcance	17
1.5. Justificación	18
2. Marco Conceptual: Aspectos de Electrofisiología Cardíaca	19
2.1. El corazón humano	19
2.2. Estructura	19
2.3. Funcionamiento	22
2.4. Sistema eléctrico del corazón	22
2.5. Variabilidad de frecuencia cardiaca (VFC)	24
2.6. Índices de dominio temporal	26
2.7. Transformada de Fourier	27
2.8. Índices de dominio frecuencial	29
2.8.1. Alta frecuencia (HF, por sus siglas en inglés)	29
2.8.2. Baja frecuencia (LF, por sus siglas en inglés)	29
2.9. Holter cardíaco	30
2.10. Electrocardiograma (ECG)	30
2.11. Tipos de ECG	30
3. Marco conceptual: Tecnologías	32
3.1. Aplicaciones web	32
3.2. MVC	32
3.3. Tecnología del lado del cliente	33
3.4. Tecnología del lado del servidor	35

3.4.1.	Bases de datos	36
3.4.2.	Base de datos relacionales	36
3.4.3.	Lenguaje SQL	37
3.5.	Lenguaje de aplicación	38
3.6.	Metodologías ágiles	40
3.6.1.	Historia	40
3.6.2.	Manifiesto ágil	40
3.7.	Scrum	41
3.7.1.	Valores de Scrum	41
3.7.2.	Proceso	42
3.7.3.	Historias de usuario	44
3.7.4.	Product backlog	45
3.7.5.	Roles	45
4.	Marco metodológico	47
4.1.	Metodología	47
4.2.	Reunión inicial	47
4.3.	Etapas de desarrollo (Iteraciones)	48
4.3.1.	Iteración 1: Diseño de la base de datos y tema visual	48
4.3.2.	Iteración 2: Instalación y configuración de los programas y bibliotecas a utilizar	49
4.3.3.	Iteración 3: Manejo de pacientes y datos aleatorios de prueba	50
4.3.4.	Iteración 4: Búsqueda y ordenamiento de los pacientes	50
4.3.5.	Iteración 5: Diagnósticos	51
4.3.6.	Iteración 6: Lectura de datos para el electrocardiograma	51
4.3.7.	Iteración 7: Lista de registros cardíacos	51
4.3.8.	Iteración 8: Instalación y configuración de las bibliotecas gráficas	52
4.3.9.	Iteración 9: Generación de estimadores estadísticos	52
4.3.10.	Iteración 10: Más estimadores estadísticos	52
4.3.11.	Iteración 11: Estimadores frecuenciales	52
4.3.12.	Iteración 12: Múltiples pacientes	52
4.3.13.	Iteración 13: Reporte	53
5.	Resultados	54
5.1.	Pruebas	54
5.2.	Iteraciones	54
5.2.1.	Iteración 1: Diseño de la base de datos y tema visual	54
5.2.2.	Iteración 2: Instalación y configuración de los programas y bibliotecas a utilizar	56
5.2.3.	Iteración 3: Manejo de pacientes y datos aleatorios de prueba	58
5.2.4.	Iteración 4: Búsqueda y ordenamiento de los pacientes	61

5.2.5. Iteración 5: Diagnósticos	61
5.2.6. Iteración 6: Lectura de datos para el electrocardiograma	63
5.2.7. Iteración 7: Lista de registros cardíacos	63
5.2.8. Iteración 8: Instalación y configuración de las bibliotecas gráficas . .	64
5.2.9. Iteración 9: Generación de estimadores estadísticos	64
5.2.10. Iteración 10: Más estimadores estadísticos	65
5.2.11. Iteración 11: Estimadores frecuenciales	68
5.2.12. Iteración 12: Múltiples pacientes	69
5.2.13. Iteración 13: Reporte	70
6. Conclusiones	73
6.1. Recomendaciones	75
Referencias Bibliográficas	76

Índice de figuras

1.1. Etapas del proyecto de desarrollo de una plataforma para un holter libre . . .	16
2.1. Capas del corazón	20
2.2. Arterias coronarias	21
2.3. El corazón y sus cavidades	21
2.4. Sistema eléctrico del corazón	23
2.5. Potenciales de acción y el ECG	24
2.6. Ondas del registro cardíaco	25
2.7. Intervalos RR	25
2.8. Ejemplo de una señal transformada del dominio del tiempo al dominio frecuencial	28
3.1. Tabla persona (en la que por ahora tenemos un sólo registro)	36
3.2. Tabla vehículo	37
3.3. Ejemplo de consulta en SQL	38
3.4. Proceso Scrum	43
5.1. Modelo de datos	55
5.2. Pantalla de ingreso de usuario	57
5.3. Interfaz de administración	57
5.4. Lista de pacientes	58
5.5. Edición de los datos del paciente	59
5.6. Ficha del paciente (datos personales)	59
5.7. Ficha del paciente (datos clínicos)	60
5.8. Búsqueda avanzada	61
5.9. Lista de diagnósticos	62
5.10. Nuevo diagnóstico	62
5.11. Registros cardíacos	63
5.12. Formulario de estadísticas del registro cardíaco/Serie temporal RR	64
5.13. Media de los segmentos de la serie temporal RR	65
5.14. Todas las gráficas del dominio del tiempo	66
5.15. Histograma	67
5.16. Estimadores de altas y bajas frecuencias para registros de un paciente	68

5.17. Formulario para las gráficas con múltiples pacientes	69
5.18. Media calculada para dos pacientes	69
5.19. Reporte en pdf del registro del paciente, página 1	70
5.20. Reporte en pdf del registro del paciente, página 2	71
5.21. Reporte en pdf del registro del paciente, página 3	72

Lista de tablas

4.1. Iteraciones o etapas de desarrollo 48

Introducción

El corazón humano es uno de los órganos más importantes de nuestro cuerpo. Si bien su función primordial es bombear la sangre a todo nuestro organismo también es un indicador del estado de éste. Gracias a esto nace la rama de la medicina conocida como cardiología, la cual se especializa en el estudio del corazón y sus enfermedades, tanto su diagnóstico como su tratamiento. Los cardiólogos se basan en muchos indicadores distintos para llevar a cabo esta tarea, como ejemplo de uno de ellos tenemos los intervalos RR, los cuales indican la duración de los latidos basándose específicamente en un tipo de onda llamada “R” y la duración entre sus apariciones consecutivas.

La cardiología siempre se ha apoyado de la tecnología para poder realizar sus estudios y avances. Desde la invención del electrocardiograma a finales del siglo XIX ha ido evolucionando cada vez más. Hoy en día tenemos monitores cardíacos y holters que registran en tiempo real la actividad eléctrica del corazón y le permite al especialista conocer su estado para de esta manera detectar y diagnosticar cualquier anomalía que se pueda presentar. En el mercado existe una gran variedad de estos dispositivos, en su mayoría cuentan con software propio cuyos datos generalmente son cerrados e inaccesibles por terceros. Si bien esto podría ser una medida de seguridad importante, es un inconveniente si se desea utilizar esos datos para investigaciones o algún otro paquete de software que utilice y/o analice estos datos.

Por ello en la Universidad Central de Venezuela se ha avanzado en un proyecto para diseñar un dispositivo totalmente abierto tanto en hardware como en software con la finalidad de que los datos puedan ser exportados con facilidad y ser utilizados por otras plataformas. Esto se está desarrollando con la asesoría de la Sección de Cardiología Experimental del Instituto de Medicina Tropical de la UCV. El proyecto consta de un holter ensamblado con hardware Arduino y placas desarrolladas con la colaboración del personal docente e investigador de la Escuela de Física - Facultad de Ciencias de la UCV y de la Escuela de Ingeniería Eléctrica - Departamento de Sistemas y Automática - Facultad de Ingeniería de la UC. A su vez se cuenta con un software visualizador escrito en C utilizando las librerías QT y un detector de intervalos RR. Por último se desea llevar a cabo el manejo de los datos de los pacientes y sus registros cardíacos.

Por ello en el presente documento se describe la propuesta e implementación de un sistema para que los especialistas puedan manejar de una forma eficiente y eficaz las historias de sus pacientes, acompañado además de una evaluación estadística con reporte

analítico y gráfico de los registros. De esta manera se facilita el manejo de los datos para el estudio y seguimiento de los registros de los pacientes y futuras investigaciones en el área. Como bien se ha mencionado, el sistema será abierto y desarrollado con herramientas de software libre que permita su escalabilidad futura y compatibilidad entre distintos sistemas en caso de ser necesario.

En el primer capítulo se tratan los objetivos de este trabajo de grado, tanto los generales como específicos, y se describe el problema a solucionar. En el segundo capítulo abarcamos los aspectos de cardiología, que abarca desde la definición y funcionamiento del corazón, hasta temas más específicos como la variabilidad de la frecuencia cardíaca y sus estimadores estadísticos temporales y frecuenciales. Luego continuamos en el capítulo tres en el que se describen las tecnologías utilizadas para implementar la solución al problema, tales como tecnología del lado del servidor, del cliente (navegador web en nuestro caso) y bases de datos, con sus especificaciones; también incluimos un apartado de la metodología Scrum utilizada en este proyecto. En el cuarto capítulo se puede encontrar la metodología, o en otras palabras, los pasos a seguir para desarrollar el proyecto, basándose en la metodología Scrum descrita en el capítulo anterior. Y por último tenemos el capítulo cinco o de resultados, donde siguiendo el mismo orden del capítulo cuatro se muestran los resultados de la aplicación de la metodología acompañados de capturas de pantalla para guiar mejor al lector. Finalizamos con un breve capítulo de conclusiones y recomendaciones, en el que se dejan las impresiones finales del proyecto y sugerencias para la continuación del trabajo.

Capítulo 1

Planteamiento del problema

En este capítulo se presenta el planteamiento del problema, se describen los objetivos y el alcance esperado en el presente trabajo de investigación.

1.1. Planteamiento del Problema

Actualmente en la Sección de Cardiología Experimental del Instituto de Medicina Tropical UCV se utilizan *holters* comerciales marca *Excorde*, los cuales sirven para monitorizar el ritmo cardíaco de los pacientes durante un período que generalmente dura un máximo de 24 horas. Este sofisticado equipo se usa para recopilar datos del ritmo cardíaco de los pacientes que acuden diariamente al instituto, dichos datos son usados tanto para diagnóstico como para investigaciones en el área.

Estos dispositivos traen consigo algunos inconvenientes, entre los que podemos nombrar: un alto coste de mantenimiento y un software propietario que no sigue estándares abiertos en cuanto al almacenamiento de sus datos; por ende no se pueden utilizar de manera libre sino dentro de la propia aplicación. Tal situación conlleva a que la información recolectada de los *holters* no se pueda extraer y utilizar en una plataforma de terceros, ya sea para apoyo en el diagnóstico, o para generar más estadísticas y funcionalidades adicionales a las del *software* original que proporciona el fabricante.

Adicionalmente la plataforma tampoco soporta una alta cantidad de información en su base de datos interna, por lo que colapsa cuando la cantidad de pacientes y registros cardíacos es muy alta; el *software* se torna lento y pesado lo cual dificulta el trabajo de los que laboran con esos equipos. La imposibilidad de extracción de datos sumado a la falta de un estándar en el software lleva a la difícil decisión de tener que eliminar los datos antiguos para de esta manera seguir utilizando el sistema; por lo que no se puede llevar un histórico para realizar estudios.

Por ello se está llevando a cabo un proyecto que utiliza tanto *hardware* libre (arduino) como *software* libre, cuya finalidad es crear un equipo de bajo costo y que a su vez se pueda expandir con más funcionalidades. Hasta el momento se ha desarrollado un *holter*

ensamblado en arduino y cuyo *firmware* está programado en herramientas de *software* libre. Adicional a esto se están agregando algoritmos de detección de anomalías como fibrilación auricular y taquicardia ventricular para de esta forma servir de soporte al médico especialista a la hora de realizar el diagnóstico.

Anteriormente se han desarrollado otros trabajos, comenzando por los trabajos de E. Álvarez, J. Jiménez, F. Moleiro y A. Rodríguez [1][2], en los años 2008 y 2010 en los cuales se sentaron las bases del proyecto, luego el punto de partida fue la tesis de doctorado de Esteban Álvarez que estudia la predictibilidad de las arritmias cardíacas [3], que derivó en otro trabajo que consistió en la realización de un prototipo, dicho prototipo fue el resultado de una tesis de maestría [4]. También se realizó otro trabajo sobre los estimadores estadísticos de riesgo cardíaco, la cual estudiaba los estimadores temporales y frecuenciales del ritmo cardíaco [6]. La detección de los latidos del corazón en un electrocardiograma era necesaria, por la que se llevó a cabo otro trabajo de grado que aparte involucró la detección de anomalías por medio del reconocimiento de patrones [5], de los datos de este último se alimentaría el presente proyecto, puesto que se trabajará con los intervalos entre latidos.

Pero más allá del desarrollo actual no existe una plataforma que facilite la interacción del médico especialista con los datos que provee el *holter* y los que genera la detección, no puede observar estadísticas ni hacer búsquedas sobre los datos de una manera fácil, además de que existe la carencia de un registro de pacientes y por ende no se lleva sus historias médicas ni un seguimiento del diagnóstico del especialista.

Tampoco se cuenta con un método de acceso a la información de manera remota. Como por ejemplo la web, lo cual facilitaría aún más el trabajo del especialista al tener que evitar trasladarse hasta dónde están los datos. Esto agilizaría el proceso de diagnosis y sería un punto importante para estar a la vanguardia de la tecnología actual.

Los las etapas del proyecto del desarrollo del holter libre lo podemos visualizar mejor en la figura 1.1, el presente trabajo está marcado en rojo.

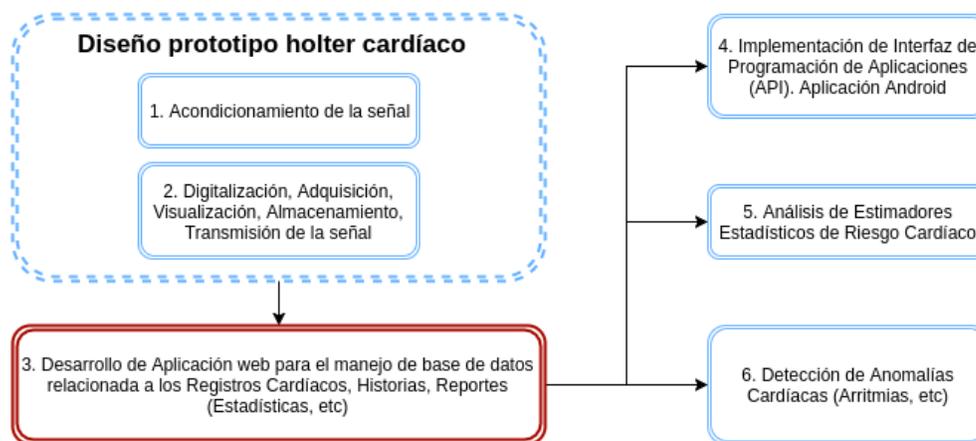


Figura 1.1: Etapas del proyecto de desarrollo de una plataforma para un holter libre

En resumen, el sistema carece de un sistema abierto y estándares libres para interactuar con el personal médico o de investigación y que posea cierta escalabilidad en el tiempo, para de esta manera tener un desarrollo continuo de nuevas funcionalidades, creación de nuevos datos y estadísticas que permitan facilitar cada vez más las investigaciones en el área.

1.2. Objetivo General

Desarrollar una aplicación web con una base de datos de señales electrocardiográficas para su visualización y estudio.

1.3. Objetivos Específicos

1. Investigar sobre los diferentes aspectos de cardiología: registro de electrocardiograma, historias clínicas, variabilidad de frecuencia cardíaca.
2. Recopilar requerimientos de los especialistas de la Sección de Cardiología del Instituto de Medicina Tropical de la UCV.
3. Seleccionar las tecnologías a utilizar y la metodología de desarrollo tomando en cuenta aspectos relacionados a las bases de datos, tecnologías del lado del servidor, tecnologías del lado del cliente (HTML, CSS, Javascript y sus bibliotecas) y los requerimientos del usuario.
4. Diseñar e implementar una base de datos que permita llevar un registro de los pacientes, incluyendo sus historias clínicas.
5. Implementar una aplicación web que considere los principales aspectos de usabilidad: consulta y administración de los datos de los pacientes.
6. Diseñar e implementar el historial de ECG (Electrocardiograma) de los pacientes con sus respectivas estadísticas (media, desviación estándar, mapa de retorno, PNN50, RMSS, LH/HF, etc).

1.4. Alcance

Este trabajo es parte de un proyecto más grande. Como se ha mencionado anteriormente se ha fabricado un holter de bajo costo usando herramientas libres, la finalidad del presente trabajo de grado es cubrir la parte del manejo de pacientes y la visualización de gráficos estadísticos. Entre las características principales del producto final, tendremos un *software* abierto (o libre), de fácil mantenimiento y escalabilidad, y con la facilidad de poder exportar sus datos para su utilización en otras plataformas más especializadas.

Si bien la aplicación tendrá un manejo de pacientes e incluye la historia médica, no se centrará en este aspecto, pero el diseño de la misma permitirá que se pueda escalar la aplicación.

1.5. Justificación

Actualmente en la Sección de Cardiología Experimental del Instituto de Medicina Tropical — UCV se tiene una plataforma costosa de mantener, con los datos cerrados y una escalabilidad prácticamente inexistente; por lo que se requiere una alternativa de bajo costo y que se adapte a los tiempos actuales.

En el instituto de cardiología el volumen de pacientes va en aumento, y dado que a cada uno se le debe realizar seguimiento es necesario poder mantener sus datos por un tiempo prolongado, para ello se creó base de datos que permite un gran número de registros que perduren en el tiempo.

Por ello se propone un producto que cubra estas necesidades, y con la intención de que se siga desarrollando y escalando aún después de finalizado el trabajo de grado.

Capítulo 2

Marco Conceptual: Aspectos de Electrofisiología Cardíaca

En este capítulo se explican los aspectos básicos relacionados con la cardiología, para ser más precisos, con la electrofisiología cardíaca: estructura y funcionamiento del musculo cardíaco, electrocardiograma (ECG), intervalos RR, indicadores temporales y frecuenciales de la variabilidad de la frecuencia cardiaca, entre otros.

Esta base conceptual es importante para entender los aspectos del presente trabajo de grado.

2.1. El corazón humano

El corazón es un órgano central del sistema cardiovascular que se encarga de bombear la sangre por todo el organismo, esto con la finalidad de proveer nutrientes y oxígeno al resto del cuerpo.[7]

Tiene un tamaño aproximado al puño cerrado de la persona, está formado por un tipo de músculo especial (músculo cardíaco) el cual funciona de manera involuntaria y está protegido por la cara torácica. El corazón suele variar su ritmo de acuerdo a diversas señales nerviosas según el esfuerzo del organismo.

2.2. Estructura

El corazón está conformado por tres capas principales (figura 2.1) como se indica en[9][10]:

- Pericardio: es la capa exterior que envuelve al corazón, y que a su vez está formado por dos capas:
 - Pericardio seroso, es el saco doble, capa más cercana al corazón, una está conformada por tejido adiposo y nervioso (pericardio visceral o epicardio) y la otra
-

es más interna y fibrosa que tiene contacto directo con el corazón (pericardio parietal).

- Pericardio fibroso: es la capa más externa, está situada por encima del pericardio seroso y se fusiona indisolublemente con el pericardio parietal.
- Miocardio: es la capa más gruesa del corazón y es el denominado músculo cardíaco, el cual se encarga de bombear la sangre. Las células musculares que lo conforman tienen la propiedad de contraerse y relajarse. Son capaces de desarrollarse en exceso (hipertrofia) en caso de que algún esfuerzo lo requiera, como por ejemplo al hacer ejercicio seguido o en caso de hipertensión arterial. En estas situaciones las paredes del corazón terminan engrosándose.

El miocardio es más ancho y grueso por el lado del ventrículo izquierdo y más estrecho por el lado de las aurículas, esto es debido a que este es el encargado de enviar la sangre oxigenada proveniente de los pulmones al resto del cuerpo y por ende requiere mucha más fuerza para impulsarla.

- Endocardio: es una capa fina que recubre las cavidades del corazón y las válvulas cardíacas por dentro, esta membrana está en contacto directo con la sangre por lo cual también cumple una función endocrina que consiste en segregar la hormona endocardin, la cual se encarga de prolongar la contracción del miocardio.

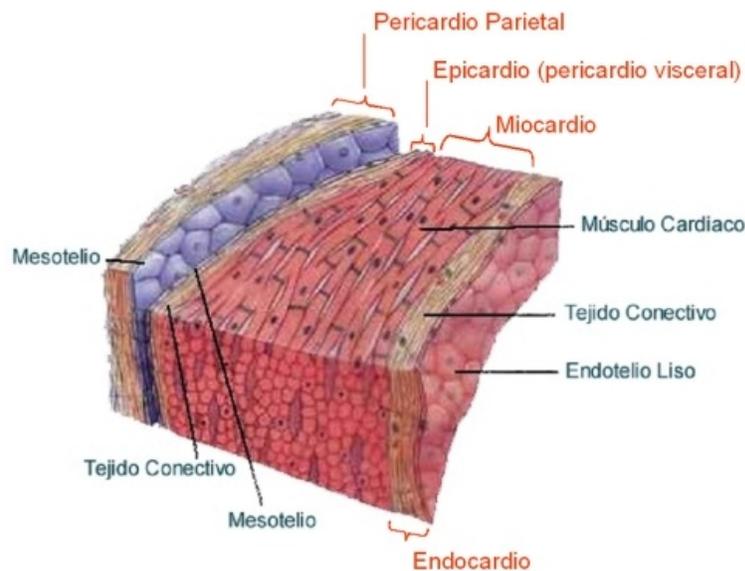


Figura 2.1: Capas del corazón

Al igual que cualquier otro órgano del cuerpo humano el corazón requiere sangre para obtener nutrientes y oxigenarse, por lo que tiene su propio sistema vascular, a este

se le denomina *circulación coronaria*; y está conformado por dos ramificaciones de la *aorta* denominadas *arterias coronarias* (figura 2.2). Una de estas ramificaciones va al lado derecho del corazón, y es relativamente pequeña debido a que esta zona es más reducida y requiere menos esfuerzo ya que sólo envía sangre a los pulmones. La otra ramificación naturalmente va hacia el lado izquierdo, y suele ser más grande dado que de este lado tenemos más masa muscular; y se requiere más esfuerzo por ser el responsable de enviar sangre al resto del cuerpo.[11]

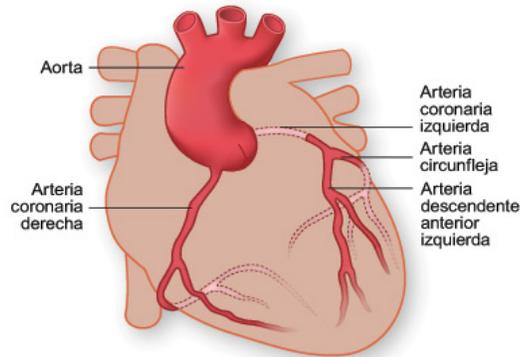


Figura 2.2: Arterias coronarias

En cuanto a su distribución, el corazón está formado principalmente por 4 cavidades (figura 2.3). Las dos superiores se llaman aurícula derecha y aurícula izquierda respectivamente, las inferiores son los ventrículos derecho e izquierdo. Cada sección del corazón es una bomba independiente, las cuales están divididas por un tabique muscular llamado septo.[8]

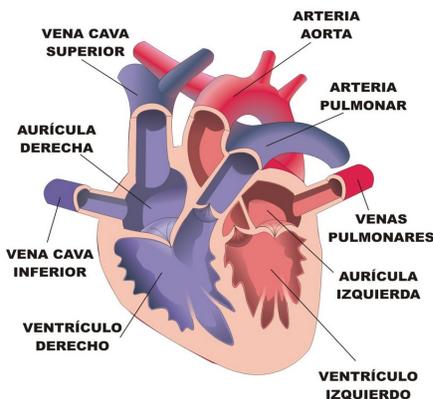


Figura 2.3: El corazón y sus cavidades

También posee cuatro válvulas por las cuales se conectan las cavidades, arterias y venas principales. Tenemos la válvula tricúspide que comunica la aurícula derecha con el ventrículo derecho; la válvula pulmonar que comunica el ventrículo izquierdo con la arteria pulmonar; la válvula mitral que comunica la aurícula izquierda con el ventrículo izquierdo y por último la válvula aórtica que comunica el ventrículo izquierdo con la aorta. Cabe destacar que el lado izquierdo y el lado derecho no se comunican entre sí.

2.3. Funcionamiento

El lado derecho del corazón recibe la sangre deficiente en oxígeno proveniente del resto del cuerpo a través de las venas cava superior e inferior, llega a la aurícula derecha, para luego pasar al ventrículo derecho pasando por la válvula tricúspide, luego se bombea hacia los pulmones pasando por la válvula pulmonar y finalmente sale por la arteria pulmonar (ver figura 2.3).[8]

El lado izquierdo en cambio recibe la sangre rica en oxígeno proveniente de los pulmones por medio de las venas pulmonares; llegando a la aurícula izquierda para luego pasar al ventrículo izquierdo a través de la válvula mitral, para finalmente ser bombeado hacia el resto del cuerpo saliendo por la válvula aórtica y llegando a la aorta.

Estos dos procedimientos funcionan en perfecta sincronía, y para ello ocurren una serie de movimientos producidos por impulsos eléctricos del sistema nervioso[8]:

- **Sístole auricular:** cuando las aurículas se contraen y envían la sangre hacia los ventrículos.
- **Sístole ventricular:** la contracción de los ventrículos, lo cual ocasiona que se envíe la sangre ya sea hacia el organismo o hacia los pulmones según sea el caso.
- **Diástole:** se refiere a la relajación del músculo cardíaco, esto con la finalidad de que llegue nueva sangre a los ventrículos y así repetir el proceso.

En resumen el lado izquierdo y el lado derecho tienen funciones claramente diferenciadas, el lado izquierdo recibe la sangre oxigenada y la envía a todo el cuerpo, y el lado derecho recibe la sangre baja en oxígeno proveniente del cuerpo y la envía a los pulmones. Como el lado izquierdo necesita enviar la sangre más lejos, tiene una mayor masa del miocardio para poder cumplir tal propósito.

2.4. Sistema eléctrico del corazón

El corazón genera por sí mismo el impulso eléctrico con el cual se contrae para bombear la sangre. En condiciones normales este impulso se autogenera desde una zona especializada de neuronas llamada nodo sinusal, este se halla ubicado en la aurícula derecha bajo

la desembocadura de la vena cava. Este nodo actúa a manera de marcapasos activándose a intervalos regulares con un ritmo de 50 a 70 latidos por minuto en un corazón sano en reposo. En la figura 2.4 podemos ver la distribución del sistema eléctrico.[12]

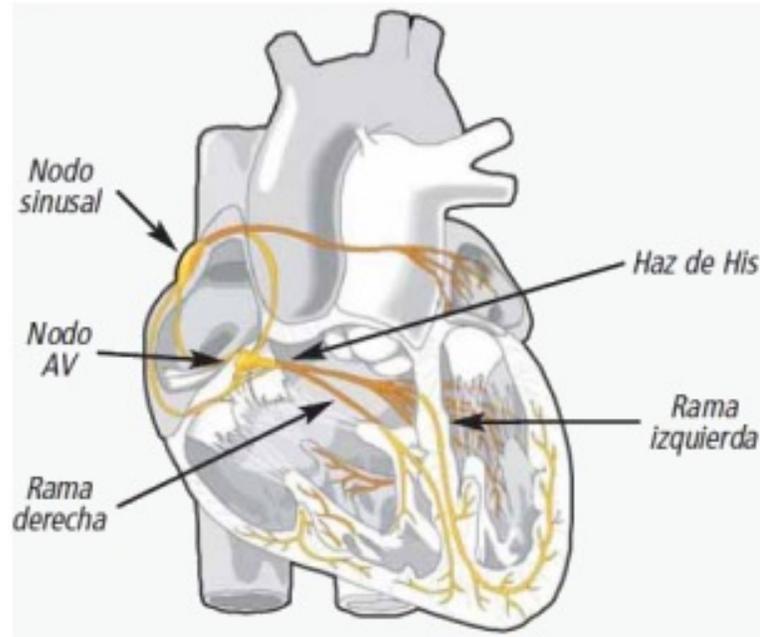


Figura 2.4: Sistema eléctrico del corazón

El proceso de contracción y relajación del músculo cardíaco se lleva a cabo gracias a la despolarización y repolarización de este nodo, el cual produce un fuerte patrón de voltaje. Esto es lo que mide los electrodos que se colocan en la piel y basándose en estos impulsos se genera un electrocardiograma.

Cuando se genera el impulso denominado potencial de acción desde el nodo sinusal, este se propaga por las dos aurículas a través de unos haces de conducción preferencial para el impulso eléctrico, para llegar finalmente al nodo AV (auriculoventricular), este nodo funciona como un centro de comunicaciones que regula el paso del impulso eléctrico hasta los ventrículos, en los cuales se generan las contracciones para el bombeo de la sangre. Los ventrículos y el nodo AV se conectan por medio de un diminuto cable denominado *haz de His*, el cual se bifurca con el fin de llevar el impulso a ambos ventrículos. De este modo el impulso eléctrico comienza en las aurículas activándose estas de primero y luego los ventrículos, luego el nodo AV retiene la descarga eléctrica a modo de condensador; hasta que las aurículas se hayan contraído por completo y los ventrículos se hayan llenado de sangre durante la fase llamada diástole. En la figura 2.5 podemos apreciar como según la activación eléctrica de las distintas áreas del corazón, se generan las ondas que conforman un latido, esto es lo que podemos apreciar en un EGC. Tal como se indica en [13][14][15][16]

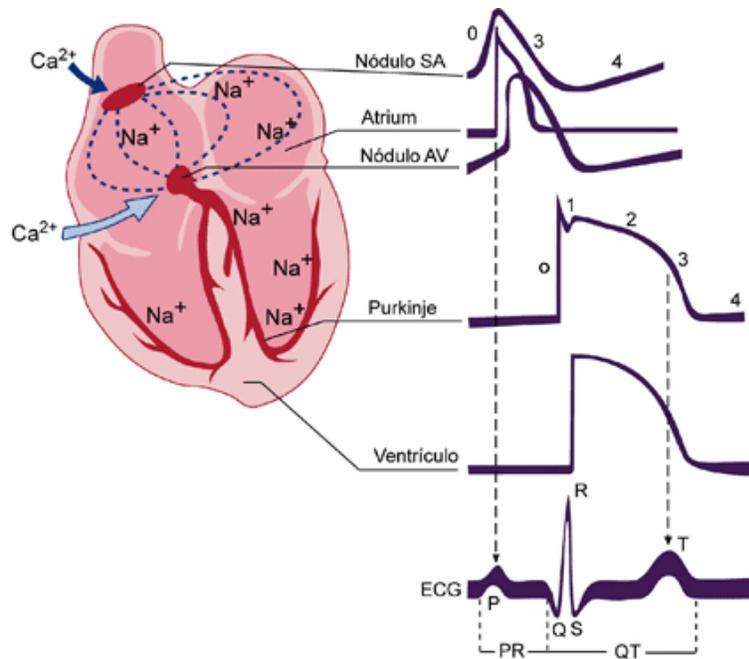


Figura 2.5: Potenciales de acción y el ECG

2.5. Variabilidad de frecuencia cardíaca (VFC)

Como se ha explicado anteriormente, los latidos parten desde el nodo sinusal, para activar los ventrículos y de esta manera bombear la sangre a los pulmones y al resto del cuerpo. Para medir esto se hace a partir de un registro electrocardiográfico, en estos registros se pueden apreciar unas ondas de un tamaño notablemente mayor a las demás, estas son las ondas R (ver figura 2.6), el intervalo entre estas ondas indica un latido y se le denomina intervalo RR (figura 2.7). Al número de latidos por minuto es a lo que llamamos frecuencia cardíaca.[17]

La frecuencia cardíaca es uno de los parámetros no invasivos utilizados para medir la actividad del corazón y conocer su estatus. En una persona sana, el tiempo entre los latidos suele variar en milésimas de segundo. La variabilidad de la frecuencia cardíaca es el estudio de estas variaciones en un intervalo de tiempo no mayor a 24 horas, con la finalidad de detectar anomalías.

Estas variaciones y/o fluctuaciones están controladas por el sistema nervioso autónomo (SNA), por lo que están condicionadas por los distintos procesos asociados a los sistemas simpáticos y parasimpáticos. La actividad cardíaca en general, incluyendo la frecuencia cardíaca y la presión arterial, se haya influenciada por diversos sistemas fisiológicos que interactúan con el sistema nervioso autónomo, tales como: el sistema respiratorio, sistema nervioso central, por medio de los estados emocionales, esfuerzo físico y mental, el sistema vaso-motor, el cual regula la contracción y relajación de los vasos sanguíneos, etc.

Gracias a que casi todo lo que ocurre en el organismo influye en la VFC, esta se ha

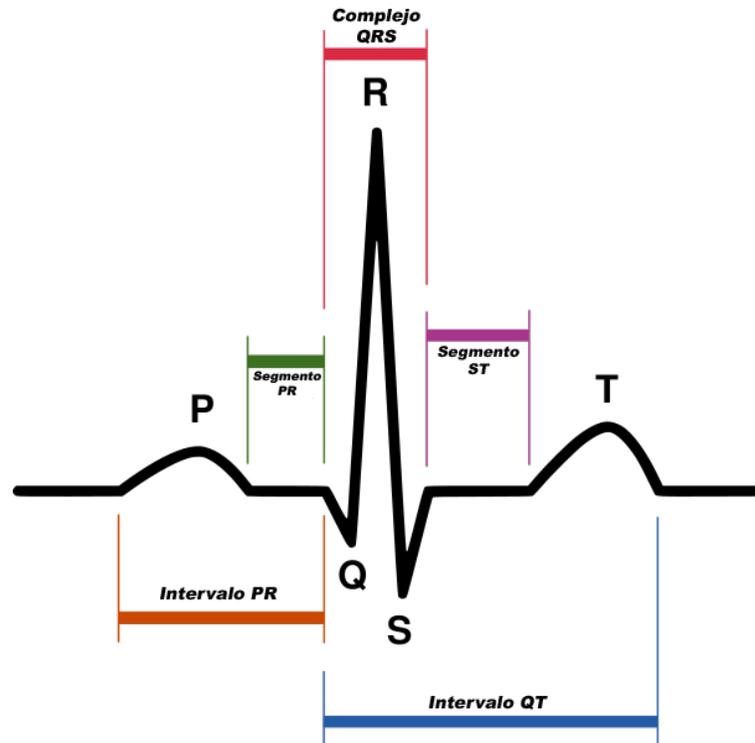


Figura 2.6: Ondas del registro cardíaco



Figura 2.7: Intervalos RR

convertido en un excelente indicador de la condición física del sujeto y su estado de salud. La manera común de medir esta variabilidad es por medio del electrocardiograma (ECG), esto se hace identificando las ondas R y midiendo los intervalos RR previamente mencionados, posteriormente estudiamos el comportamiento dinámico que presentan va-

rios indicadores o índices estadísticos sobre esta serie de intervalos RR. Algunos de estos indicadores serán expuestos a continuación.

2.6. Índices de dominio temporal

Como se indica en [36], los índices del dominio temporal más utilizados son:

- **Media RR.** Como su nombre lo indica, es el promedio de la duración de los intervalos RR en un período de tiempo o en todo el registro.

$$Media_{RR} = \frac{1}{n} \sum_{i=1}^n RR_i,$$

donde n es el número de intervalos temporales RR

- **SDNN.** Indica la desviación estándar con respecto al valor medio previamente mencionado, al igual que este, puede ser por un período de tiempo o en todo el registro.

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2},$$

donde n es el número de intervalos temporales RR,

x_i = intervalo temporal RR,

μ = valor medio del registro

- **SDNNindex.** Se refiere al promedio de todas las desviaciones estándar obtenidas en períodos cortos de tiempo los cuales generalmente son de 5 minutos.

$$SDNNindex = \frac{1}{m} \sum_{i=1}^m sd_i,$$

m = número de períodos cortos de tiempo o segmentos,

sd_i = desviación estándar del período corto de tiempo o segmento.

- **SDANN.** Es la desviación estándar de los promedios de los intervalos RR en períodos cortos que generalmente son de 5 minutos. Este indicador es sensible a los cambios de posición del paciente y a los cambios de la actividad cardíaca en el ciclo circadiano.

$$SDANN = sd(\text{med}S_1, \dots, \text{med}S_m),$$

S_i = segmento de cinco minutos.

- **pNN50.** Es una medida porcentual de la variación entre latidos cardíacos consecutivos que tienen una diferencia mayor a 50 ms. Se utiliza como indicador de los cambios rápidos en la frecuencia cardíaca.

$$pNN50 = \frac{\text{contador}}{n - 1} * 100 \%,$$

n es el número de intervalos RR,

El *contador* representa la cantidad de intervalos RR adyacentes con una diferencia mayor a 50 ms.

- **rMSSD.** Representa la raíz cuadrática media de las diferencias sucesivas de los intervalos RR vecinos (adyacentes). Este índice al igual que el pNN50 registra los cambios rápidos en la variabilidad de la frecuencia cardíaca, debido a que mide los cambios entre latidos y provee un indicador del control cardíaco vagal (tono parasimpático), pues bien refleja la integridad del nervio vago mediada por el control autónomico del corazón.

$$rMSSD = (\langle (\Delta RR_i)^2 \rangle)^{1/2}$$

2.7. Transformada de Fourier

Toda señal (en este caso también aplica para la serie temporal RR) sin importar su complejidad, se puede descomponer en una suma de funciones periódicas de distintas frecuencias. La transformada de Fourier es un proceso que nos permite obtener los coeficientes de las funciones sinusoidales que forman la señal original. Esto es, la señal representada por una sumatoria de señales sinusoidales.[37]

Este proceso nos permite transformar la señal en el dominio del tiempo al dominio frecuencial, permitiéndonos de este modo estudiar sus componentes, la información contenida en la señal sin embargo no se altera, sólo se representa de una manera distinta, siendo posible aplicar el proceso a la inversa. La razón de este cambio se debe a que el mundo frecuencial aporta información de una forma mucho más clara con la que se pueden obtener o eliminar características relevantes de la señal tales como las altas o bajas frecuencias, información de gran interés para nuestro estudio. En la figura 2.8 podemos ver un ejemplo de esta transformación.

La Serie de Fourier es de utilidad para representar ciertas funciones. Las funciones periódicas en el dominio de la frecuencia están descritas por las series de Fourier, es decir, por la frecuencia fundamental y sus armónicos. Una función se considera periódica cuando la señal en cuestión repite sus valores cada cierto intervalo de la variable independiente.[38]

Las señales no periódicas son de dos clases: cuasi-periódicas y transitorias. Las cuasi-periódicas no son periódicas en el sentido matemático pero tienen una descripción discreta

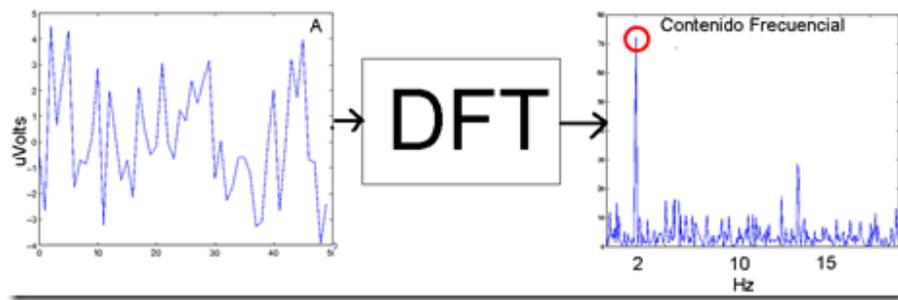


Figura 2.8: Ejemplo de una señal transformada del dominio del tiempo al dominio frecuencial

en el dominio de la frecuencia, que difiere de la anterior en que los términos que aparecen no son armónicos de alguna frecuencia fundamental. Una combinación de señales periódicas no correlacionadas dan lugar a una cuasi-periódica. Las transitorias serían las que no cumplen las condiciones anteriores.

La actividad de bombeo del corazón proviene de un sistema intrínseco de conducción eléctrica, que consiste en la conjunción de varias ondas representadas en un único esfuerzo muscular, y la señal que describe dicho comportamiento es de tipo no periódica, para señales no periódicas $x(t)$ definidas en la recta real, la herramienta que mide la cantidad que la señal tiene de frecuencia es la *transformada de Fourier*. [39][41]

La transformada de Fourier de $x(t)$ para un espacio continuo se define como

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-i2\pi ft} dt, \quad (2.1)$$

donde $e^{i\alpha} = \cos \alpha + i \sin \alpha$

Las señales que se obtienen midiendo cantidades a intervalos igualmente separados en el tiempo producen un valor para cada medición, se llaman señales discretas, esta es la forma en que se trabaja en un computador, puesto que no tenemos manera de representar infinitos puntos. De manera análoga a como se define la transformada de Fourier de una función continua, la transformada de Fourier discreta (TFD) de la señal f es:

$$f[k] = \sum_{n=0}^{N-1} f[n] e^{-\frac{2\pi i k n}{N}}, \quad k = 0, 1, 2, \dots, N-1 \quad (2.2)$$

Esta ecuación produce otra sucesión de N valores que indica la cantidad que tiene la señal de frecuencia k . El cálculo directo de la TFD con la fórmula 2.2 requiere $2 * N^2$ operaciones (N^2 multiplicaciones y N^2 sumas). La transformada de Fourier rápida (TFR) es un algoritmo que reduce el orden de complejidad del cálculo de la TFD a $4 * N \log_2 N$ operaciones haciendo una reordenación de los cálculos. [43]

La TFR es uno de algoritmos más usados en cálculos con señales unidimensionales. En señales continuas, donde el cálculo de la transformada de Fourier a partir de la ecua-

ción 2.1 puede resultar complicado, el valor de la señal se muestrea a intervalos igualmente espaciados y se usa la TFR. Si el número de muestras N es grande la percepción visual de la señal reconstruida con estos valores es similar a la señal original y en muchas casos indistinguible para el ojo humano.[42]

Una de las principales utilidades en cardiología al moverse del dominio temporal al frecuencial es poder diferenciar los componentes de altas y bajas frecuencias, las altas frecuencias indican la influencia del sistema parasimpático (el cual disminuye el ritmo cardíaco) del organismo sobre el corazón, y las bajas frecuencias indican la actividad del sistema simpático (el cual aumenta el ritmo cardíaco). Con estos datos el especialista puede comprobar varios aspectos, entre ellos el estado general del organismo, monitorear como se comporta el corazón durante las horas de sueño (en el cual la actividad parasimpática y simpática se turnan para mantener el corazón activo, pero no demasiado), entre otros.[40]

2.8. Índices de dominio frecuencial

Los índices en el dominio frecuencial explora las propiedades de las series de intervalos RR en el espacio de Fourier, de tal manera que se pueda obtener información sobre los componentes frecuenciales presentes en ellos. Estos están relacionados con los sistemas fisiológicos que interactúan con el *sistema nervioso autónomo*; los cambios en sistemas como el respiratorio resultan en cambios en la VFC. Para ser precisos, los cambios en frecuencia están estrechamente relacionados con los sistemas simpático y parasimpáticos.[36]

2.8.1. Alta frecuencia (HF, por sus siglas en inglés)

Contempla las bandas de frecuencias entre 0,15 y 0,43 Hz, la alta frecuencia está relacionada con los cambios rápidos en la actividad cardíaca, debido a que está ligado al sistema parasimpático, esto incluye también los cambios en el sistema respiratorio los cuales están en las bandas comprendidas entre 0,2 y 0,35 Hz.

2.8.2. Baja frecuencia (LF, por sus siglas en inglés)

Representa mayormente la actividad del sistema simpático, sus bandas de frecuencia están entre 0,04 y 0,15 Hz, refleja la regulación de la presión sanguínea. Dado que esta banda es más compleja, también suele mostrar la interacción de ambas ramas del sistema nervioso autónomo.

Para casos más puntuales se suele tomar en cuenta también la muy baja frecuencia (VLF, por sus siglas en inglés) que indica los cambios lentos en la actividad cardíaca, y la ultra baja frecuencia (ULF, por sus siglas en inglés) relacionado con el ciclo circadiano.

2.9. Holter cardíaco

Un holter es un dispositivo utilizado para monitorizar la actividad eléctrica del corazón durante un período de tiempo. También se le conoce como monitor *holter* o *holter-ecg*.

Una de las características principales de un *holter* es que es portátil, ya que su uso está pensado para monitorizar la actividad cardíaca mientras el paciente desarrolla sus actividades cotidianas. Por ello los *holter* generalmente se les colocan a los pacientes por un período de 24 horas y de esta manera se obtiene el registro completo de la actividad en el ciclo circadiano.[18][19]

Su nombre se debe a su inventor Norman Holter, un físico estadounidense de principios del siglo XX especializado en la rama de la biofísica. El primer holter era un monitor cardíaco ligero del tamaño de un maletín fácilmente transportable. Antes de esto los monitores ocupaban una habitación exclusiva del hospital y requerían mucho tiempo y preparación previa. Luego a medida que la tecnología fue avanzando los holter se fueron volviendo más compactos, a tal punto que ya tienen el tamaño de un teléfono móvil.

Con este dispositivo no sólo se puede medir la actividad electrocardiográfica, sino también la frecuencia cardíaca y la respiratoria, los cuales se ven reflejados en el electrocardiograma.

2.10. Electrocardiograma (ECG)

Es una prueba diagnóstica que evalúa el ritmo y la función cardíaca a partir de un registro de la actividad eléctrica del corazón. Los impulsos eléctricos normalmente se expresan en una cuadrícula (generalmente en papel milimetrado cuando es impreso) en forma de líneas o curvas, las cuales traducen la contracción o relajación de las aurículas y ventrículos en distintos tipos de onda.

El ECG se puede realizar tanto en un consultorio como en un hospital, o incluso puede ser generado por un holter durante las actividades normales del paciente. Todo depende de la condición del mismo y lo que el especialista requiera evaluar.

Según la naturaleza y el comportamiento de las ondas, el médico puede determinar si el paciente tiene una anomalía, e incluso el ECG es tan representativo que permite detectar que tipo de anomalía es; tal como un ataque cardíaco previo, corazón agrandado, arritmias, etc.[20]

2.11. Tipos de ECG

- ECG estándar, el cual se realiza en estado de reposo y en un consultorio.
 - ECG de esfuerzo, se realiza mientras se hace ejercicio. Tiene como finalidad mostrar cómo funciona el corazón en estado de estrés. Tiene varias utilidades, desde diag-
-

nosticar cardiopatía coronaria hasta determinar cuánto ejercicio puede realizar el paciente tras una cirugía o ataque cardíaco.

- ECG de 24 horas, el cual se lleva a cabo por medio de un Holter y muestra la actividad cardíaca durante un día y una noche. En casos muy contados, se puede usar por más de 24 horas. En algunos casos se le indica al paciente llevar un registro de sus actividades o anotar si tiene algún síntoma.

Con estos conceptos podemos tener un mayor entendimiento de la propuesta y ejecución de la misma que se desarrolla en este documento. Los indicadores, tanto del dominio temporal como del frecuencial, son puntos que veremos en funcionamiento en este trabajo de grado, por ello es importante conocerlos y tener una idea básica de su funcionamiento.

Capítulo 3

Marco conceptual: Tecnologías

En este capítulo se explican las herramientas tecnológicas utilizadas en la implementación del presente proyecto. Se cubren conceptos claves como aplicaciones web y el patrón MVC, para ir profundizando luego con las tecnologías del lado del servidor (Bases de datos, Django como aplicación de servidor, etc) y las del lado del cliente o navegador (HTML, Bootstrap, CSS, Javascript); y finalizar con la descripción del método de trabajo a utilizar, en este caso *Scrum*.

3.1. Aplicaciones web

Se refiere a las aplicaciones que se acceden desde un navegador, en muchos casos los datos que se muestran en la misma se han generado desde un servidor.

Este tipo de arquitectura se denomina cliente-servidor y consiste básicamente en que el cliente hace un requerimiento y el servidor consulta sus datos, los procesa y le devuelve los resultados de acuerdo a los parámetros dados por el cliente.

Cuando hablamos de aplicaciones web el cliente se refiere al navegador, el cual es el que interactúa directamente con el usuario. El cliente a su vez provee una interfaz en HTML, CSS y Javascript, lo cual conforma una página web y funciona como medio visual con el usuario para la interacción con el sistema. Por otro lado tenemos al servidor, el cual almacena los datos y provee la lógica de negocios, el cual contempla las funcionalidades clave del sistema tales como consulta a la base de datos, validaciones de los datos que llegan del cliente, cálculos estadísticos, entre otros.[21]

3.2. MVC

Conocido como Modelo-Vista-Controlador, es una propuesta de ingeniería del software para la implementación de sistemas que requieran interfaces de usuario. Esta metodología surge de la necesidad de hacer las aplicaciones más organizadas y escalables, donde cada uno de los elementos estén desacoplados de la mayor forma posible.[22]

El MVC es un modelo en tres capas, como su nombre sugiere estas son:

- **Modelo:** es la capa que maneja la interacción de la aplicación con la base de datos, acá se generan las consultas y generalmente se convierte los datos a objetos u otro tipo de estructuras del lenguaje utilizado para la aplicación web. Gracias a esto se abstrae completamente la base de datos del resto de la aplicación. La principal ventaja es que podría cambiarse la base de datos (por ejemplo, de *MySQL* a *PostgreSQL*) sin tener que modificar poco o nada del resto de la aplicación.
- **Vista:** como su nombre lo indica, representa la parte visual de la aplicación, para ser más concretos, la tecnología del lado del cliente (navegador o ventanas de la aplicación por citar dos ejemplos), la función de la capa de la vista es mostrar resultados. En el caso web, involucra tecnologías tales como *HTML*, *CSS*, *Javascript* y sus librerías como *JQuery* o *Bootstrap*.
- **Controlador:** es el “*cerebro*” de la aplicación, se encarga de la lógica de negocios de la misma y contiene los cálculos necesarios y la preparación de los datos recibidos desde el modelo, para que sean recibidos por la vista y sean mostrados al usuario. En cierto modo representa la capa intermedia entre el modelo y la vista.

En la actualidad el Modelo-Vista-Controlador viene implementado en los diversos frameworks web modernos. Sin embargo este modelo no se limita a este campo, se puede utilizar en cualquier tipo de aplicación que requiera acceso a datos y una presentación de los mismos, incluso en aplicaciones móviles.

3.3. Tecnología del lado del cliente

Como bien se ha mencionado, el cliente provee la interacción con el usuario y hace uso de tres tecnologías mencionadas previamente:

- **HTML.** Son las siglas de *Hiper Text Markup Languaje*, no es un lenguaje de programación, sino como su nombre en inglés lo sugiere, es un lenguaje de marcado, en el cual lo que se hace básicamente es delimitar sectores de la página. Este lenguaje se utiliza para crear el esqueleto de la página web, contiene desde los textos que lo conforman, hasta los *metadatos* que la describen (algunos de estos utilizados por los motores de búsqueda de Internet para conocer datos claves del contenido de las páginas). Sin embargo, aunque un HTML por si solo puede presentar información de manera legible, en muchos casos no ofrece una interfaz amigable por sí sola, por lo que necesita ayuda de los estilos CSS y las funcionalidades Javascript.[23]
 - **CSS.** Siglas de *Cascade Style Sheets*, u hojas de estilos de cascada, este es un lenguaje utilizado para darle formato al HTML, ya esto contempla la parte visual del
-

sitio web en cuestión. En un archivo CSS por lo general se hace referencia a secciones del HTML, de esta forma se le da el formato a esa sección o texto. La gran ventaja de los CSS es que no sólo nos permite mostrar una interfaz agradable, sino que al estar separado del HTML permite cambiar el estilo manteniendo los contenidos y la estructura de la página intactos o con mínimas modificaciones, siempre y cuando se hayan llevado buenas prácticas a la hora de maquetar la página.[24]

- **Javascript.** Es un lenguaje de programación para navegadores. Gracias a este lenguaje se pueden definir comportamientos, como por ejemplo al dar click sobre un menú hacer que este despliegue sus opciones. Gracias al poder de Javascript las páginas web tienen la capacidad de proveer una interacción más intuitiva con los usuarios, en las tendencias actuales se ha logrado minimizar la diferencia entre una aplicación web y una de escritorio gracias a las herramientas que nos provee este lenguaje.

Debido al avance de estas tres tecnologías actualmente las aplicaciones web han llegado a una interacción muy natural con el usuario y ha permitido que el desarrollo de aplicaciones se traslade a la web, lo cual acoplado con el Internet nos trae grandes ventajas, como el poder acceder datos a distancia y sin tener que instalar nada aparte de un navegador web.[25]

Sin embargo, para un desarrollo más ágil y estándar con las tecnologías mencionadas, se han creado bibliotecas y actualizado a lo largo de varios años con la finalidad de abstraerse de tareas repetitivas y comunes, así el desarrollador se dedica enteramente a la lógica del negocio, evitando de esta manera lo que llamamos "*la reinvencción de la rueda*".

- **Bootstrap.** Es una biblioteca que provee módulos que frecuentemente se utilizan en las páginas web, como por ejemplo menús o tablas. Esta biblioteca está hecha para trabajar con HTML5 y CSS, por lo que provee estructuras que simplemente se colocan en el archivo fuente y ya viene con sus estilos predefinidos, de esta manera se evita tener que escribir el HTML y CSS correspondiente y apenas hacer ligeras modificaciones. Bootstrap también incluye una biblioteca Javascript la cual provee los comportamientos.[26]
 - **JQuery.** Como se mencionó en *Bootstrap*, este trae consigo bibliotecas Javascript, sin embargo esta a su vez utiliza unas librerías llamadas *JQuery*, la cual provee funcionalidades muy útiles como manipulación del HTML (por ejemplo, hacer aparecer un texto), animaciones, interacción asíncrona con el servidor gracias a la tecnología AJAX, manipulación del CSS (por ejemplo, cambiar de color un texto), etc.
 - **AJAX.** Son las siglas de *Asynchronous JavaScript And XML*, es una tecnología del navegador que permite cargar datos desde el servidor sin tener que recargar la página, su gran ventaja es que el tráfico de datos es mucho menor por no tener que
-

cargar el HTML, estilos e imágenes nuevamente y la interacción con la página se hace más rápida e incluso intuitiva.

En sus comienzos AJAX utilizaba principalmente XML (*eXtensible Markup Language*) como formato para transportar los datos al servidor y viceversa, sin embargo, la tendencia actual es utilizar un formato más sencillo llamado JSON (*Javascript Object Notation*), el cual es el formato de los objetos en javascript y el navegador lo puede interpretar de manera nativa, es decir, sin tener que utilizar bibliotecas adicionales.[27]

Esta tecnología la podemos ver en funcionamiento en aplicaciones web famosas como *Gmail*, *Twitter*, *Facebook*, entre otras.

3.4. Tecnología del lado del servidor

La tecnología del lado del servidor suele variar, mientras en el cliente (navegador) siempre tenemos HTML, CSS y Javascript (siendo en realidad estos dos últimos, opcionales), y con variantes en las bibliotecas, del lado del servidor varía desde el sistema operativo hasta el lenguaje del formato de salida, incluyendo la base de datos y el lenguaje/interprete bajo el que se implementa la lógica de negocios. Habitualmente por razones de estabilidad se utilizan servidores en sistemas operativos basados en *Unix*, por lo general *Linux*.

En cuanto al almacén de los datos, este puede variar desde ser un simple sistema de archivos, que sería una carpeta en el sistema, hasta bases de datos relacionales o no relacionales. Todo depende de lo que se desea implementar, por ejemplo, si la aplicación es simplemente una página informativa, no necesitamos almacenar más datos que los archivos HTML, CSS, Javascript e imágenes dentro de nuestro sistema de archivos; en cambio si deseamos realizar un sistema con control de usuarios, que se pueda alimentar desde el cliente, etc, se utilizan bases de datos.

Por último, para implementar la lógica de negocios hay miles de alternativas y generalmente se segmentan según el lenguaje de programación a utilizar, tenemos aplicaciones implementadas en *PHP*, *Python*, *Java*, *Ruby*, *Perl*, e incluso binarios escritos en *C*, a su vez existen bibliotecas y *frameworks* para cada uno de estos lenguajes.

Es importante tener clara la definición de un *framework*, como su nombre lo sugiere, es un marco de trabajo, el cual provee metodologías para desarrollar y la organización de la aplicación, esto con la finalidad de estandarizar el desarrollo y permitir que su mantenimiento y escalabilidad sea más llevadera a futuro. Además de proveer metodologías, un framework generalmente tiene un conjunto de bibliotecas con funcionalidades comunes como, por ejemplo, el manejo de formularios y consultas a las bases de datos.

La elección del lenguaje y su framework como del soporte de datos se hace de acuerdo a los requerimientos del sistema que se desea desarrollar. Para nuestro caso, el soporte de datos sería un manejador de bases de datos relacional, y el lenguaje a utilizar sería *Python* con su conocido *framework Django*.

Cédula	Nombre	Fecha de nacimiento	Estatura
<u>1234567</u>	<u>Alvaro</u> Marquina	14/05/1985	1.75

Figura 3.1: Tabla persona (en la que por ahora tenemos un sólo registro)

3.4.1. Bases de datos

Una base de datos no es más que un repositorio centralizado de datos, esta nos permite tenerlos en un solo sitio evitando así que tengamos discordancia en la información con la que estamos trabajando. Las bases de datos permiten realizar cuatro operaciones básicas: crear, modificar, eliminar y leer registros.

Pueden almacenar distintos tipos de datos, desde texto, números y fechas, hasta información binaria como imágenes y audio.

Existen varios tipos de bases de datos, pero para nuestro propósito, utilizaremos la más común: la base de datos relacional.[28]

3.4.2. Base de datos relacionales

En un principio, las bases de datos no eran más que archivos de texto cuyos campos se delimitaban con un carácter tal como una tabulación o un punto y coma. Esto implicaba un gran costo computacional a la hora de realizar búsquedas sobre los datos debido a que se realizaban de forma lineal, mientras más información más lento era su manejo.[29]

Las bases de datos relacionales nacen para resolver este problema, en vez de tener un gran archivo, los datos se organizan en tablas, para verlo de una manera más sencilla imaginemos las tablas como la representación de algo real, por ejemplo persona, podemos tener una tabla persona que contiene el nombre, fecha de nacimiento y estatura. Digamos que queremos almacenar los vehículos que tiene esa persona, por lo que creamos otra tabla llamada vehículo, donde tenemos año, placa y por supuesto, el dueño que sería una persona. Bien, como es poco práctico almacenar los datos del dueño en la misma tabla del vehículo (puesto que una persona puede tener varios vehículos y estaríamos repitiendo datos), simplemente en esta tabla se almacena una referencia a una persona en su respectiva tabla.

Como podemos observar en el ejemplo presentado en las tablas 3.1 y 3.2, en la columna dueño del vehículo placa AAA 123, en vez de colocar los datos del dueño, usamos la cédula a manera de referencia, entonces en este caso simplemente para saber el dueño del vehículo placa AAA 123, sólo buscamos a la persona con la cédula 1234567.

Las base de datos relacionales toman su nombre debido a que las distintas tablas tienen

Año	Placa	Dueño
2000	AAA 123	<u>1234567</u>

Figura 3.2: Tabla vehículo

la capacidad de relacionarse entre sí, esto gracias a referencias como en el ejemplo anterior, lo que nos permite una mejor abstracción y búsquedas de los datos de una manera más estructurada y eficiente.

Otra de las ventajas de las bases de datos relacionales, es que nos ofrecen indexación, en la cual un campo de una tabla se toma como índice, tal como en un libro, el índice no es más que una etiqueta con una referencia al registro, para mantener la analogía, el índice es el nombre del capítulo del libro, y el número de página indica donde está el registro (que sería nuestro capítulo). Esto nos provee más velocidad en las búsquedas, puesto a que igual que un libro, es más rápido buscar en un índice que hoja por hoja, en especial cuando la cantidad de datos es muy grande. Generalmente se pueden definir cuantos índices sean necesarios, ya queda a criterio del diseñador de la base de datos establecer que se necesita, todo esto acoplándose al problema a resolver.

Retomando el ejemplo de las personas y vehículos, como se va a buscar sobre la cédula, entonces esta se puede definir como un índice. Al ser un campo numérico esto nos permite optimizar las búsquedas mejor que si fuese un texto, puesto a que los números se pueden comparar directamente si son mayores o menores. Esto permite crear índices con árboles binarios, los cuales permiten búsquedas el doble de rápidas.

Los manejadores de bases de datos relacionales son paquetes de software que implementan las bases de datos relacionales. Entre los más conocidos tenemos *Oracle*, *MySQL*, *PostgreSQL* (que de hecho son servidores) y *SQLite* (usado generalmente para datos locales y tecnologías móviles).

Las bases de datos relacionales suelen por lo general utilizar un lenguaje para su manejo llamado SQL (Lenguaje de consulta estructurado por sus siglas en inglés). Este lenguaje permite realizar las cuatro operaciones básicas mencionadas anteriormente: Crear, modificar, borrar y eliminar registros. A su vez proveen sintaxis para la creación de las bases de datos, y administración de sus tablas, las cuales a su vez pueden ser creadas, modificadas, léidas y eliminadas.

3.4.3. Lenguaje SQL

Sus siglas en español significan Lenguaje de Consulta Estructurado, se utiliza para las consultas en las bases de datos y permiten definir criterios de búsqueda los cuales pueden

```
SELECT vehículo.* FROM vehículo JOIN persona ON vehículo.dueño = persona.cédula WHERE persona.estatura = X
```

Figura 3.3: Ejemplo de consulta en SQL

incluir hasta las relaciones con otras tablas.

En el ejemplo de las personas y vehículos podríamos querer todos los vehículos que pertenezcan a las personas con una estatura X, si nos fijamos, en la tabla vehículo (3.2) no tenemos la estatura de la persona, sino su cédula, por lo que aprovechamos las ventajas que nos provee SQL para poder obtener este dato. Existen varias maneras de hacerlo, por ejemplo la que mostramos en la figura 3.3.

En este ejemplo la consulta se puede interpretar de la siguiente manera: selecciona todos los campos de la tabla vehículo (el * indica que son todos los campos según el estándar), de la tabla vehículo (3.2) unida a la tabla persona (3.1) por medio de los campos dueño y cédula (que son los campos con los que se relacionan ambas tablas) donde la persona tenga la estatura X.

Si bien SQL es un lenguaje estándar para las consultas, suele tener pequeñas variaciones según el manejador, e incluso el conjunto de instrucciones suele variar un poco, por esto no siempre una consulta escrita para el manejador *Oracle* podría ejecutarse igual en el manejador *PostgreSQL*.

3.5. Lenguaje de aplicación

- **Python.** Es un lenguaje interpretado, interactivo y orientado a objetos, el cual incorpora módulos, excepciones, tipado dinámico y clases. Su nombre proviene del famoso grupo de comediantes ingleses *Monty Python* y fue creado por Guido Van Rossum a finales de los 80 y principios de los 90.[30]

Este lenguaje combina una sintaxis limpia con un poder extraordinario, ya que contiene interfaces para llamadas al sistema y se puede extender por medio de código en *C* o *C++*. También es bastante portable gracias a sus intérpretes para los diversos sistemas operativos que tenemos en la actualidad tales como los basados en *Unix* (*Linux*, *BSD*, o incluso el famoso *Mac*), *Windows*, o hasta intérpretes para algunos dispositivos móviles.

Entre sus grandes bondades tenemos la facilidad de aprendizaje, y que gracias a su sencilla sintaxis contamos con estructura que abstraen acciones comunes como por ejemplo recorrido de listas, parámetros variables, etc, y que gracias a esto nos permite un desarrollo más ágil de nuestras aplicaciones.

Gracias a todo esto *Python* se ha vuelto bastante popular y se utiliza en aplicaciones famosas como *Dropbox* y muchos de los desarrollos de *Google*.

- **Django.** Es un framework web de alto nivel en *Python* que fomenta desarrollar aplicaciones de manera rápida, limpia y práctica.[31]

Entre sus características más destacables tenemos:

- ORM integrado: (Mapeo objeto relacional por sus siglas en inglés) *Django* provee clases para la interacción con la base de datos. A partir de estas clases se crean (por medio de la herencia) los modelos que se corresponden directamente con las tablas. Su poderosa biblioteca aprovecha las bondades de la sintaxis de *Python* para hacer casi cualquier tipo de consulta sin tener que escribir código *SQL* ya que este lo genera. Sin embargo, en caso de ser necesario también se pueden escribir directamente estas consultas en contados casos.
- Control de acceso: el *framework* ya trae consigo el control de usuarios, provee incluso scripts genéricos para la creación y manejo de usuarios y sus roles. Para esto implementa su propio control de sesiones y permisos por cada vista de manera individual.
- Interfaz de administración: aunque su uso es opcional, *Django* incluye un módulo de administración que nos permite manejar el contenido de la base de datos de nuestra aplicación web. Incluyendo la información asociada a los usuarios.
- URLs elegantes: o si se les puede llamar de otra manera: URLs amigables, las cuales pueden ser configuradas a gusto del desarrollador.
- Sistema de plantillas: *Django* provee un poderoso sistema de plantillas totalmente desacoplado del resto de la aplicación, de esta manera permite mantener separada la lógica de negocios de la parte visual. La gran ventaja que esto ofrece es que si queremos rediseñar el estilo de la página no tenemos que reescribir todo sino únicamente las plantillas.
- Sistema caché: las vistas o páginas no siempre cambian, por lo que para optimizar la velocidad de carga se implementa una caché que guarda las vistas ya procesadas.
- Internacionalización: Nos provee la capacidad de multilinguaje en nuestras aplicaciones sin tener que escribir código para cada idioma. Sólo variando los textos.

Entre las aplicaciones famosas escritas con *django* tenemos *Disqus* (sistema de comentarios para páginas web), *Instagram* (web social de fotografías móviles), el sitio de *Mozilla* y *Pinterest* (pizarra virtual donde los usuarios comparten contenido que encuentran en la web).

3.6. Metodologías ágiles

3.6.1. Historia

En los años 90 surgieron varios movimientos denominados metodologías livianas (*Light-weight methodologies*); entre estas tenemos *Extreme programming* (XP), Scrum, Software Craftmanship, Lean Software Development, etc.[32][34]

En febrero del año 2001, se reunieron en Utah (EEUU) un grupo de diecisiete profesionales reconocidos del desarrollo del software y referentes de las tecnologías livianas del momento, con el objetivo de determinar los valores y principios que les permitirían a los equipos desarrollar software de forma más acertada con las necesidades del cliente, y responder mejor a los cambios que pudieran surgir a lo largo de un proyecto de desarrollo. La finalidad era elaborar una alternativa a las metodologías tradicionales, los cuales se caracterizaban por la rigidez y estar dominados por la documentación.

El resultado de esta reunión fue la *Agile Alliance*, una organización sin fines de lucro cuya finalidad es promover los valores y principios de la filosofía ágil y su adopción en la industria. Acá también surgió la piedra angular del movimiento ágil: el manifiesto ágil.

3.6.2. Manifiesto ágil

El manifiesto ágil se sustenta bajo cuatro valores extraídos directamente de su página [33]:

Estamos descubriendo formas mejores de desarrollar software tanto por nuestra propia experiencia como ayudando a terceros. A través de este trabajo hemos aprendido a valorar:

Individuos e interacciones sobre procesos y herramientas

Software funcionando sobre documentación extensiva

Colaboración con el cliente sobre negociación contractual

Respuesta ante el cambio sobre seguir un plan

Esto es, aunque valoramos los elementos de la derecha, valoramos más los de la izquierda.

El manifiesto ágil establece las bases de las metodologías ágiles por medio de doce principios que mencionamos directamente desde el manifiesto

- Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
 - Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
-

- Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.
- Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
- Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
- El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
- El software funcionando es la medida principal de progreso.
- Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.
- La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.
- La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
- Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
- A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

3.7. Scrum

Scrum es una metodología ágil en la cual se aplican un conjunto de buenas prácticas para obtener el mejor resultado posible de un proyecto, está enfocado en las entregas parciales y regulares del producto final. Scrum está indicado para proyectos en los que se necesita ver resultados pronto, donde los requisitos pueden variar o no están completamente definidos, y que la innovación, competitividad, flexibilidad y productividad son fundamentales.

3.7.1. Valores de Scrum

Para trabajar en Scrum se necesita una base firme de valores que sirvan como fundamento para el proceso y los principios del equipo. A través del uso del trabajo en equipo y la mejora continua, Scrum tanto crea como depende de estos valores. Éstos son Foco, Coraje, Apertura, Compromiso y Respeto.

- **Foco.** Porque nos enfocamos en sólo unas pocas cosas a la vez, trabajamos bien juntos y producimos un resultado excelente. De este modo logramos entregar ítems valiosos antes.
- **Coraje.** Porque no estamos solos, nos sentimos apoyados y tenemos más recursos a nuestra disposición. Esto nos da el coraje para enfrentar desafíos más grandes.
- **Apertura.** Durante el trabajo en conjunto expresamos cotidianamente cómo nos va y que problemas encontramos. Aprendemos que es bueno manifestar las preocupaciones, para que éstas puedan ser tomadas en cuenta
- **Compromiso.** Porque tenemos gran control sobre nuestro destino, nos comprometemos más al éxito.
- **Respeto.** A medida que trabajamos juntos, compartiendo éxitos y fracasos, llegamos a respetarnos los unos a los otros, y a ayudarnos mutuamente a convertirnos en merecedores de respeto.

3.7.2. Proceso

Scrum se ejecuta en bloques llamados iteraciones, las cuales generalmente duran entre una y dos semanas, aunque en algunos equipos pueden durar más, dependiendo del proyecto y el equipo de desarrollo. Cada iteración debe proporcionar un resultado completo, el cual representa un incremento en el producto final que será entregado al cliente.

Al principio hay una lista de objetivos o requisitos priorizados del producto, el cual funciona como plan del proyecto. En esta lista el cliente establece prioridades de acuerdo al valor que aportan de acuerdo a su coste, ya sea en tiempo como monetario; estas tareas serán repartidas a lo largo de todas las iteraciones.

En la iteración se llevan a cabo distintas actividades que se definen a continuación:

Planificación

El primer día de la iteración se realiza una reunión de planificación que se realiza en dos partes

- **Selección de requisitos (4 horas máximo)** El cliente presenta al equipo la lista de requisitos priorizada del producto o proyecto. El equipo pregunta al cliente las dudas que surgen y selecciona los requisitos más prioritarios que se compromete a completar en la iteración, de manera que puedan ser entregados si el cliente lo solicita.
 - **Planificación de la iteración (4 horas máximo)** El equipo elabora la lista de tareas de la iteración necesarias para desarrollar los requisitos a que se ha comprometido. La estimación de esfuerzo se hace de manera conjunta y los miembros del equipo se autoasignan las tareas.
-

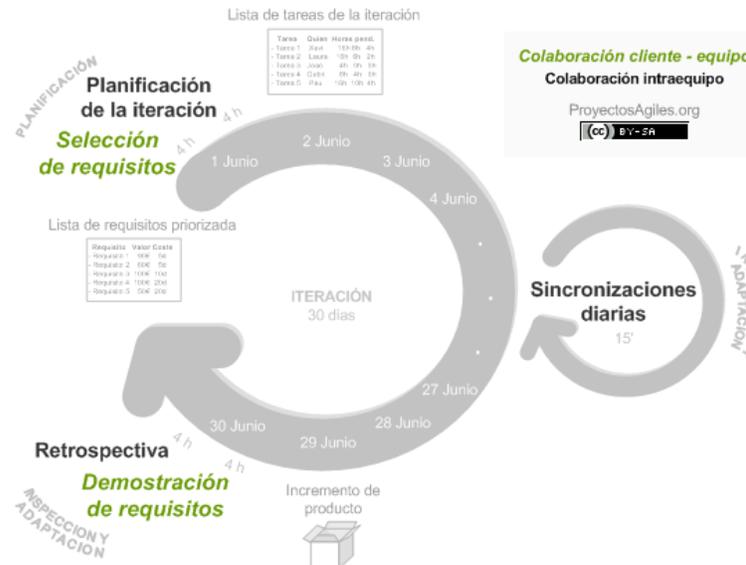


Figura 3.4: Proceso Scrum

Ejecución de la iteración

Diariamente el equipo realiza una reunión de sincronización de quince minutos como máximo. Cada miembro del equipo inspecciona el trabajo del resto, tomando en cuenta aspectos como dependencia entre tareas, el progreso hacia el objetivo de la iteración y los obstáculos que se puedan presentar. Para llevar a cabo esta tarea cada miembro del equipo debe responder a tres preguntas:

- ¿Qué he hecho desde la última reunión de sincronización?
- ¿Qué voy a hacer a partir de este momento?
- ¿Qué impedimentos tengo o voy a tener?

Durante la iteración el *Facilitador (Scrum Master)* se encarga de que el equipo pueda cumplir con los objetivos y velar por la productividad. El *Scrum Master* elimina los obstáculos que el equipo no puede resolver por sí mismo; y protege al equipo de interrupciones externas que puedan afectar el cumplimiento de los objetivos.

Inspección y adaptación

El último día de la iteración se realiza la reunión de revisión; y se realiza en dos partes:

- **Demostración** (4 horas máximo). Se le presenta al cliente el producto de la iteración, se muestran los resultados en forma de incremento del producto. El cliente realiza las adaptaciones y cambios necesarios de acuerdo a estos resultados, de esta manera se obtiene un producto fiel a sus necesidades.

- **Retrospectiva** (4 horas máximo). El equipo analiza los fallos y aciertos de la iteración, logrando determinar como mejorar la productividad para las siguientes iteraciones.

3.7.3. Historias de usuario

Una historia de usuario se refiere a un requisito o funcionalidad descrita por el cliente, debe su nombre al hecho de que en vez de ser una descripción técnica suele ser descrita en forma de historia corta, incluyendo a su actor (generalmente un tipo de usuario) y la acción que desea realizar. Uno de los requisitos clave a la hora de crear una historia de usuario es que debe hacerse cara a cara con el *Product Owner*.

Las historias de usuario surgieron en *eXtreme Programming* (XP) como una respuesta a una situación habitual en los proyectos de desarrollo de software; los clientes y los especialistas de negocio usualmente se comunican con los equipos de desarrollo por medio de extensos documentos conocidos como especificaciones funcionales. Estas especificaciones a su vez están sujetas a interpretaciones distintas entre las partes involucradas, lo cual trae como consecuencia que el producto final sea muy distinto al producto esperado por el cliente.

Se componen de tres elementos principales, los cuales son conocidos como *las tres Cs* y toda historia de usuario debe tenerlos.

- **Card (Ficha)**. Toda historia de usuario debe poder describirse en una ficha de papel pequeña. Si la historia no se puede describir en ese espacio, entonces estamos transpasando las fronteras y comunicando demasiada información que debería compartirse cara a cara.
- **Conversación**. Toda historia de usuario debe ser creada junto al *Product Owner* en una conversación presencial. Esto dado que no sólo se intercambia la información, sino también pensamientos y opiniones por medio de los gestos; logrando así una comunicación más efectiva.
- **Confirmación**. Toda historia de usuario debe estar explicada de tal manera que el equipo de desarrollo entienda lo que el *Product Owner* espera que se construya. Esto se conoce también como criterios de aceptación.

La redacción de una historia de usuario generalmente se realiza personificando el rol que necesita la funcionalidad y explicando para que. Es la manera más sencilla y menos ambigua de realizarla. El formato a utilizar es el siguiente:

Como [ROL] necesito [FUNCIONALIDAD] para [BENEFICIO]

Este formato es el sugerido por Mike Cohn, uno de los colaboradores para la creación de Scrum y fundadores de la *Scrum Alliance*.

Este estilo de redacción nos trae una series de beneficios a la hora de la interpretación e implementación de las funcionalidades:

- **Primera persona.** Nos permite meternos en el rol del usuario que utilizará la funcionalidad.
- **Priorización.** Al tener esta estructura el *Product Owner* comprende mejor la historia de usuario y ayuda a priorizarla en el *Backlog*.
- **Propósito.** Al conocer el propósito de la funcionalidad permite establecer alternativas al equipo de desarrollo que permitan el mismo beneficio; esto en caso de que la funcionalidad tenga mucho costo de producción o su implementación no sea viable.

Las historias de usuario son básicamente los ítems de nuestra lista de tareas al que llamamos *Backlog* y cada una de estas contiene una estimación en puntos que indica su complejidad.

3.7.4. Product backlog

Es un artefacto esencial en Scrum. Consiste en una lista ordenada por prioridad de idea (o mejor dicho, historias de usuario) para el producto. De esta lista es que se extraen las actividades que realizará el equipo de desarrollo, y es la única fuente posible de estas tareas. Cada ítem del *Product Backlog* contiene su respectiva descripción y estimación. Cada nueva funcionalidad, cambio o *bug* debe estar en el *Product Backlog*.

Esta lista generalmente empieza siendo una lista breve e imprecisa, y a medida que se va refinando el producto también se va refinando el *Backlog*, de esta forma las historias quedan mejor definidas y las estimaciones más precisas.

El responsable de mantener el *Product Backlog* es el *Product Owner*, en muchos casos con ayuda del resto de equipo, ya sea para crearlo como para refinarlo.

3.7.5. Roles

- ***Product Owner*.** Es el responsable de delimitar el producto con el máximo valor posible para la fecha deseada. El *product owner* gestiona el flujo de trabajo del equipo, seleccionando y refinando los ítems del *Product backlog*, establece las prioridades y conoce a fondo los requerimientos y el negocio. Como su nombre lo sugiere, es el responsable y conocedor de todo el producto.
 - **Miembro del equipo de desarrollo.** Se refiere a cada uno de los profesionales que llevan a cabo el trabajo necesario para el incremento del producto. Se organizan así mismos para realizar el trabajo. El *Product Owner* crea una lista de las tareas por prioridad, los desarrolladores se encargan de hacer la estimación del tiempo de cada una de las tareas.
-

- ***Scrum Master***. Se puede tomar como el "líder del proyecto"; se encarga de apoyar al resto del equipo para realizar sus tareas. El *Scrum Master* trabaja junto al *Product Owner* y le proporciona apoyo a la hora de crear y mantener el *Product Backlog*. A su vez, trabaja junto al equipo de desarrollo para la implementación de las funcionalidades que permitirán obtener un incremento del producto al finalizar la iteración. El *Scrum Master* se encarga de eliminar los obstáculos que impidan el avance del equipo, ya sean internos o externos.

El *Scrum Master* a su vez es como una especie de *coach* para el equipo, por lo que requiere tener altos conocimientos en la metodología *Scrum*; ayuda a los miembros de equipo a ejecutar y aprender la metodología.

Capítulo 4

Marco metodológico

En este capítulo se describe la metodología que se llevó a cabo para cumplir los objetivos, es decir, el desarrollo de la aplicación.

4.1. Metodología

Para el presente proyecto se tomaron en cuenta distintas variantes para determinar que metodología era acorde con el desarrollo.

Entre estos aspectos tenemos:

- La frecuencia con las que se harían las reuniones, esto se determinó basándose en las que se hicieron durante el desarrollo del seminario de tesis.
- Una planificación preliminar de que aspectos se les mostrarían a los médicos que asesoraron el proyecto basado en lo ya propuesto en el seminario.
- La necesidad de mostrar resultados y obtener un *feedback* de manera rápida y efectiva.
- A pesar de haber una propuesta clara, es muy posible que hayan cambios a pedido de los médicos, siempre tomando en cuenta que no se salgan del alcance del proyecto.
- El equipo de desarrollo es de una sola persona

Por lo tanto se tomó la decisión de trabajar utilizando una adaptación de *Scrum*, la cual es una metodología ágil que provee una gran flexibilidad y a su vez permite mostrar resultados de manera rápida.

4.2. Reunión inicial

Se realiza una reunión inicial con los especialistas de la Sección de Cardiología del Instituto de Medicina Tropical de la UCV, con la finalidad de recabar datos sobre los

requerimientos, y obtener asesoría sobre los distintos aspectos médicos que involucra el desarrollo de esta aplicación.

Luego se realiza otra reunión con los tutores, para determinar el alcance y las herramientas a utilizar. De acá surge la idea por propuesta propia de utilizar las tecnologías mencionadas en el capítulo dos. También se realiza un plan de trabajo para comenzar a desarrollar.

4.3. Etapas de desarrollo (Iteraciones)

Siguiendo el plan de trabajo se establecieron las tareas a seguir para cumplir los objetivos, estas tareas se distribuyeron a lo largo de varias iteraciones con el fin de hacer entregas parciales, todo esto siguiendo la metodología Scrum. Cada iteración fue de una o dos semanas para ajustarse a las reuniones con los médicos especialistas, después de cada reunión ya se tenían validadas las funcionalidades y una lista de correcciones a la aplicación. En la mayoría de las veces dichas correcciones se mostraban en la siguiente reunión.

Las iteraciones se llevarán a cabo en el orden indicado en el cuadro 4.1:

Iteración	Tareas
Iteración 1	Diseño de la base de datos y tema visual
Iteración 2	Instalación y configuración de los programas y bibliotecas a utilizar
Iteración 3	Manejo de pacientes y datos aleatorios de prueba
Iteración 4	Búsqueda y ordenamiento de los pacientes
Iteración 5	Diagnósticos
Iteración 6	Lectura de datos para el electrocardiograma
Iteración 7	Lista de registros cardíacos
Iteración 8	Instalación y configuración de las bibliotecas gráficas
Iteración 9	Generación de estimadores estadísticos
Iteración 10	Más estimadores estadísticos
Iteración 11	Estimadores frecuenciales
Iteración 12	Múltiples pacientes
Iteración 13	Reporte

Tabla 4.1: Iteraciones o etapas de desarrollo

4.3.1. Iteración 1: Diseño de la base de datos y tema visual

Lo primero que se realizó fue definir el modelo de datos, para ello se utilizó una conocida herramienta gráfica llamada *MySQLWorkbench*, que aunque haya sido diseñada

para trabajar con una base de datos *MySQL* y no *PostgreSQL* permite crear un esquema visual de la base de datos para guiarnos.

La segunda actividad en esta iteración fue la creación de la plantilla que tendrá la aplicación web, esto fue meramente desarrollado en HTML y CSS usando el *framework Bootstrap*.

4.3.2. Iteración 2: Instalación y configuración de los programas y bibliotecas a utilizar

Una vez validado el modelo de datos (tras una explicación a los especialistas) y haber mostrado la plantilla, se procede a instalar *Django*, *PostgreSQL* y definir el resto de las dependencias del proyecto. Se escribe el código para el manejo del modelo de datos en *Python*, y se integra la plantilla en la aplicación.

Instalación del sistema de autenticación y control de acceso para salvaguardar la confidencialidad de los datos, esto debido a que se maneja información personal de los pacientes. Configuración del sitio de administración de *Django*, esto para poder visualizar y modificar el contenido de las tablas en la base de datos.

Para el sistema de autenticación y control de acceso se definieron dos roles principales: administrador y usuario (médico, docente e investigador, etc).

- *Administrador*. Este rol tiene acceso total a todas las funciones del sistema, desde la zona de administración donde se puede modificar directamente las tablas de la base de datos, hasta la parte de pacientes. Este nivel de acceso generalmente se le provee a personal calificado en área técnica, ya que modificar directamente los datos podría corromper la integridad del sistema.

Entre las posibilidades del administrador está la creación de nuevos usuarios, la edición de sus datos o incluso deshabilitarlos, su función principal es esta, dado que no se requieren conocimientos médicos sino técnicos para cumplir sus funciones.

- *Usuario*. Se refiere al rol básico, sin privilegios directos sobre la base datos, se le asigna al médico, especialista, técnicos e investigadores, puede manejar la lista de sus pacientes, sus datos, historia, registros cardíacos y diagnósticos. Cada médico puede acceder únicamente a sus pacientes. Como la especialidad del usuario que utiliza este rol no implica conocimientos técnicos del sistema, su acceso se encuentra limitado al área descrita; el usuario no puede modificar directamente las tablas de la base de datos ni manejar otros usuarios. Esto es para evitar daños accidentales al sistema.
-

4.3.3. Iteración 3: Manejo de pacientes y datos aleatorios de prueba

- Realización de las vistas para listar a los pacientes, agregar nuevos pacientes y editar sus datos. Esto implica realizar nuevos archivos HTML, CSS y Javascript, adaptándolo a la plantilla principal creada en la primera iteración; para la lista de pacientes se crea una tabla en donde se verán sus datos principales, como nombre, cédula de identidad, edad y fecha de ingreso al sistema. Entre las funcionalidades de esta lista tenemos también un buscador básico (busca por nombre, apellido o cédula) y la capacidad de ordenar los pacientes a gusto del usuario.
- Adicional a la lista tenemos una vista de detalle, la cual muestra más datos del paciente, como lugar y fecha de nacimiento, datos clínicos y socioeconómicos, y una foto si se encuentra disponible.
- Para agregar un paciente se crea un formulario donde el usuario debe llenar los datos arriba descritos, el formulario cuenta con dos secciones: datos personales y datos clínicos. Este mismo formulario es utilizado para la edición de los datos del paciente.
- Creación de funciones de generación de datos aleatorios de prueba. Esta es una función generadora pensado para ser usado en tiempo de desarrollo, en ella se generan pacientes aleatorios para poder visualizarlos en las distintas pantallas del sistema, y así poder cerciorarse de un correcto funcionamiento del mismo.

4.3.4. Iteración 4: Búsqueda y ordenamiento de los pacientes

Los pacientes se pueden desactivar para no ser mostrados por defecto, esto sería el equivalente a borrarlos, pero por seguridad sus datos se mantienen en la base de datos, sólo pudiéndose eliminar definitivamente desde la interfaz de administración.

Implementación de la función de búsqueda, ordenamiento y filtrado de los pacientes. En la iteración anterior se crearon la vistas, pero únicamente la parte visual con HTML y CSS, en esta parte se realiza la implementación; en la búsqueda se hace a dos niveles, uno básico, la cual se encuentra en la lista de pacientes y sólo busca por cédula, nombre y apellido, y una búsqueda más avanzada con filtrado, la cual permite buscar por rango de edades, nombre, apellido, cédula y filtrar por pacientes activos e inactivos. Por último se implementan funciones para el ordenamiento de la tabla, ya sea por nombre, apellido, cédula, edad o fecha de ingreso. Este ordenamiento se puede realizar tanto en la lista principal como en los resultados de búsqueda.

4.3.5. Iteración 5: Diagnósticos

Creación y listado de diagnósticos de los pacientes, se pueden agregar tantos diagnósticos como sean necesarios, de tal manera que se pueda llevar un historial.

Para la creación se implementa un formulario en el cual el especialista selecciona una anomalía y tiene un cuadro para rellenar con observaciones sobre el paciente. El listado se muestra por medio de una tabla donde indica la anomalía y la observación. Con la intención de mantener un historial de cambios, los diagnósticos no se pueden borrar del sistema.

4.3.6. Iteración 6: Lectura de datos para el electrocardiograma

Se realizó un script para leer los datos; la entrada (o señal, en nuestro caso) es un archivo de texto con los puntos del gráfico, también se realiza la lectura de otro archivo de texto con la duración de los intervalos RR. Estos últimos datos son los que se utilizarán para el cálculo de los estimadores estadísticos.

Para esta funcionalidad se debe diseñar tablas en la base de datos para contener los datos leídos. En un principio se crea una tabla llamada *registro* a la cual se le asocia una tabla aparte con los puntos que representan los intervalos temporales (RR). En la tabla *puntos* se guardan las coordenadas de la gráfica, pero adicional a esto para los registros RR se guarda la sumatoria de los valores RR desde los puntos anteriores al actual, esto con la finalidad de poder realizar búsquedas por intervalos de tiempo. Recordemos que los intervalos RR es la distancia temporal expresada en milisegundos entre latidos, al tener la suma se tiene el tiempo transcurrido desde el inicio del registro hasta ese punto.

Una vez leído los archivos y almacenados en la base de datos, ya están listos para ser gráficos y calcular sus estadísticas e indicadores.

4.3.7. Iteración 7: Lista de registros cardíacos

Creación de listas de registros cardíacos, aquí se muestran los registros de un paciente, se pueden visualizar distintos tipos de registro, ya sea electrocardiograma o intervalos RR. En esta lista se incluyen datos básicos como la duración, hora de inicio y hora de finalización.

Al dar click sobre esta lista se debe ir al formulario para visualización de datos. El usuario ingresará los parámetros para calcular la gráfica y estadísticas del registro cardíaco. Se requieren datos como tamaño del segmento a graficar, el tamaño de los intervalos dentro de este segmento, hora de inicio, hora de fin y tipo de gráfica a mostrar.

4.3.8. Iteración 8: Instalación y configuración de las bibliotecas gráficas

Instalación y configuración de la biblioteca *Matplotlib*, la cual es utilizada para generar los gráficos.

Se hacen pruebas graficando electrocardiogramas por medio del formulario, indicando el tamaño del intervalo a mostrar. También se genera el esqueleto del módulo para graficar, de tal forma que cada gráfica nueva a desarrollar se haga de una manera simple.

4.3.9. Iteración 9: Generación de estimadores estadísticos

Se empieza a generar los primeros estimadores estadísticos, se calcula la media y desviación estándar para todo el registro.

Graficación de estos estimadores dado un segmento y tamaño de intervalo por medio del formulario de visualización. Se calcula cada estimador (por ejemplo la media) por cada intervalo dentro del segmento de tiempo, cada intervalo equivale a un punto en la gráfica.

4.3.10. Iteración 10: Más estimadores estadísticos

Cálculo y graficación del resto de los estimadores temporales: SDNNindex, SDANN, pNN50, rMSSD. Estos son los utilizados por los especialistas para realizar el diagnóstico. Al igual que en la iteración anterior se hace por cada intervalo dentro de un segmento más grande de tiempo. Con esto los especialistas pueden estudiar las variaciones en el tiempo de cada uno de los estimadores antes mencionados de una manera más eficaz.

4.3.11. Iteración 11: Estimadores frecuenciales

Cálculo de los estimadores frecuenciales, utilizando la transformada de *Fourier*, se calcula las altas y bajas frecuencias, posteriormente se grafica. Para realizar este objetivo se utilizan las funciones de la conocida biblioteca llamada *numpy*, la cual está especializada en cálculos matemáticos incluyendo las series de Fourier que utilizamos en esta sección.

4.3.12. Iteración 12: Múltiples pacientes

Cálculo de los estimadores para múltiples pacientes. De la lista de pacientes se pueden seleccionar varios y realizar cálculo para ese grupo. Al seleccionarlos se va a un formulario similar al de la iteración 9, en el cual igualmente el usuario ingresa el inicio del segmento, el fin y el tamaño de los intervalos, se muestran varias gráficas en una, de esta manera el especialista puede comparar datos entre distintos pacientes, esto puede ser útil por ejemplo con fines académicos y de investigación, donde el especialista puede mostrar la diferencia entre un paciente sano y uno con problemas, como apoyo en la ejecución de un diagnóstico.

4.3.13. Iteración 13: Reporte

Creación del reporte con todas las gráficas y estimadores, en formato PDF. Para esto se utiliza la biblioteca *Reportlab*, y siguiendo las sugerencias e indicaciones de los especialistas. El reporte se basa en el informe escrito que usan en la Sección de Cardiología.

Capítulo 5

Resultados

De acuerdo al plan de trabajo propuesto en la metodología, se presentan los resultados en este capítulo, se incluyen capturas de la aplicación y sus funcionalidades ya implementadas.

5.1. Pruebas

Las pruebas se realizaron con el 100 % de los especialistas de la Sección de Cardiología del Instituto de Medicina Tropical UCV. En el presente trabajo no se incluyen pruebas finales debido a la naturaleza de la metodología, al final de cada iteración se le presentaban los resultados a los médicos, ellos los validaban, daban sus observaciones y aprobaban cada funcionalidad nueva presentada.

Los especialistas probaban la aplicación de la siguiente manera, se les daba una pequeña charla inicial para explicarles los nuevos aspectos de la aplicación, no se les mencionaba donde estaba, a medida que iban utilizando la aplicación se iba observando si tenían dificultad para manejarla, se anotaban las observaciones, al final de la prueba se les pedía una retroalimentación al respecto.

Basándonos en las pruebas se creaban correcciones para la siguiente iteración, donde se les mostraba lo modificado y lo nuevo de la aplicación.

Este método de pruebas se realizó de esta manera para acoplarlo a la metodología de desarrollo Scrum.

5.2. Iteraciones

5.2.1. Iteración 1: Diseño de la base de datos y tema visual

En la figura 5.1 podemos apreciar el modelo de datos resultante, este representa un ejemplo de las tablas que se utilizarán en la base de datos. Esta diagramación es el resultado del análisis de los requerimientos solicitados.

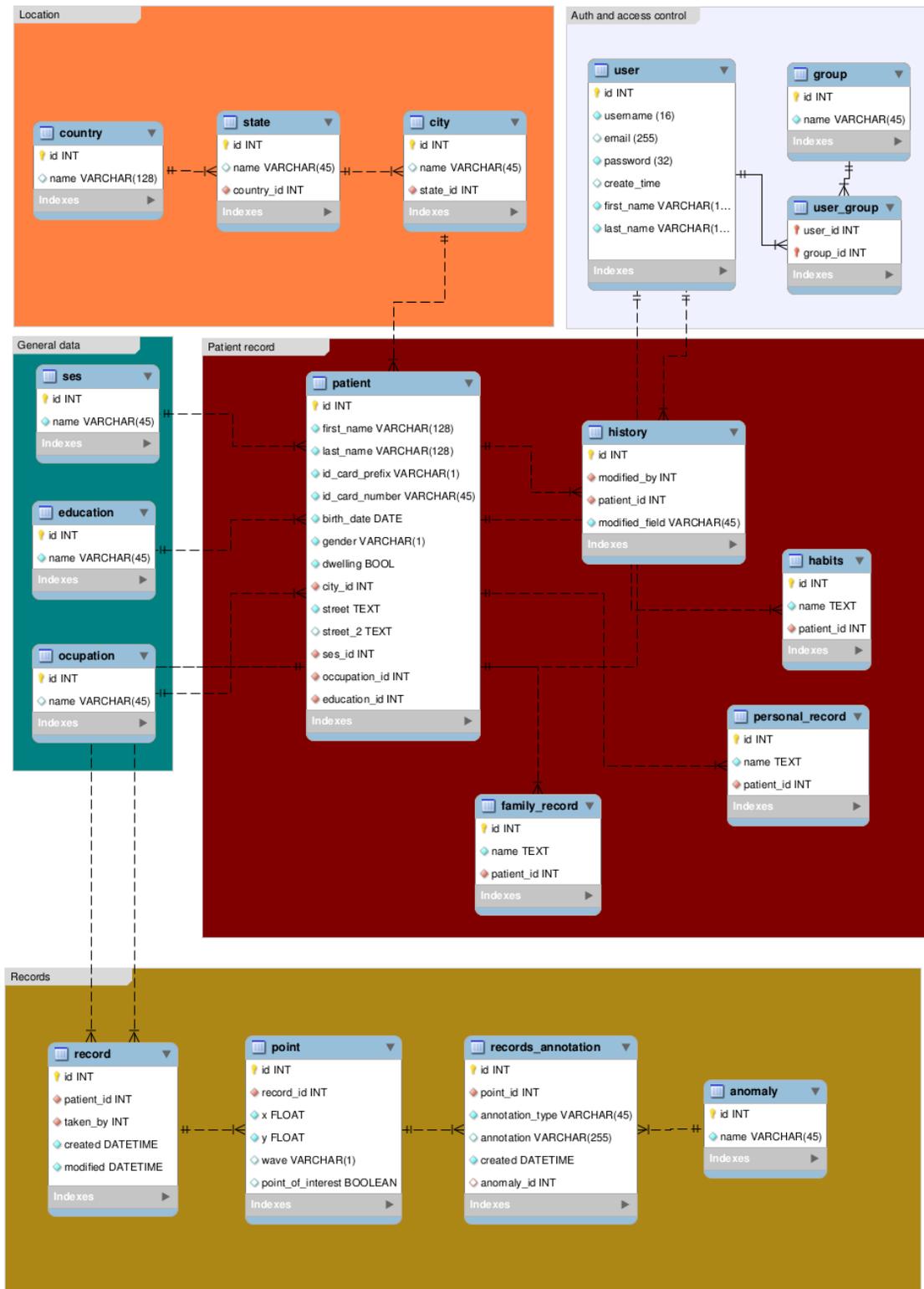


Figura 5.1: Modelo de datos

Esta base de datos fue dividida en distintas secciones en el gráfico para su fácil visualización, como se puede observar, contiene una sección de usuario, la cual incluye las tablas necesarias para el control de acceso; en nuestro caso dichas tablas ya vienen incluidas con *Django*.

Tenemos por otra parte los datos generales, que son usados para configurar los valores disponibles en distintos formularios; como por ejemplo tenemos el estatus socioeconómico (SES por sus siglas en inglés), cuyos valores podrían contener A, B, C y D, la tabla educación (*education*) la cual contiene valores como primaria, secundaria, universitaria. Esta clase de valores se incluyen en la base de datos con la finalidad de que sea posible su fácil configuración por parte del administrador del sistema.

Los datos de los pacientes los agrupamos en cinco tablas entre las cuales tenemos la tabla paciente (*patient*) donde guardamos los datos personales y clínicos, la tabla historia (*history*) en la que tendremos el registro de cambios, entre lo que se incluye que se modificó, quien lo hizo y la fecha. El resto de las tablas almacenan datos clínicos tales como hábitos personales y antecedentes.

Por último tenemos las tablas más importantes del sistema, las cuales contienen los registros cardíacos (*record*), con sus puntos (*points*), anotaciones (*record_annotation*) y anomalías (*anomaly*) detectadas en un instante de tiempo del registro.

El tema visual lo podremos ver en todas las capturas que siguen en este capítulo, fue realizado con Bootstrap 3, el diseño resultante fue una interfaz sencilla pero visualmente agradable.

5.2.2. Iteración 2: Instalación y configuración de los programas y bibliotecas a utilizar

En esta iteración se instalaron en un entorno local las bibliotecas necesarias para la realización del proyecto, principalmente *Django*. Posterior a esto se procede a instalar el sistema de autenticación del mismo e integrarlo con la plantilla, el resultado se puede ver en la figura 5.2.

Para ello se utiliza el modelo de usuario que trae consigo el *framework* y que está reflejado en el modelo de datos mostrado en la iteración anterior. Luego se integra una plantilla personalizada a la librería y tenemos el resultado mostrado.

La creación de usuarios y la asignación de privilegios se hace por medio de la interfaz de administración incluida en *Django*, podemos ver una captura en la figura 5.3. En esta sección accesible sólo por el administrador, podemos agregar y editar los registros contenidos en las tablas de la base de datos; esta acción debe realizarse con mucho cuidado porque toda modificación o borrado es irreversible.

Physiocardexp UCV

Inicio / Ingresar al sistema

Ingreso al sistema

Usuario

Contraseña

Ingresar

Realizado por Alvaro Marquina

Figura 5.2: Pantalla de ingreso de usuario

Administración de Django

BIENVENIDO/A GREGORY VER EL SITIO / CAMBIAR CONTRASEÑA / TERMINAR SESIÓN

Sitio administrativo

AUTENTICACIÓN Y AUTORIZACIÓN

Grupos	+ Añadir	Modificar
--------	----------	-----------

CITIES

Alternative names	+ Añadir	Modificar
Cities	+ Añadir	Modificar
Countries	+ Añadir	Modificar
Districts	+ Añadir	Modificar
Postal codes	+ Añadir	Modificar
Regions	+ Añadir	Modificar
Subregions	+ Añadir	Modificar

PATIENTS

Diagnósticos	+ Añadir	Modificar
Estatus socioeconómicos	+ Añadir	Modificar
Historial de cambios	+ Añadir	Modificar

Recent actions

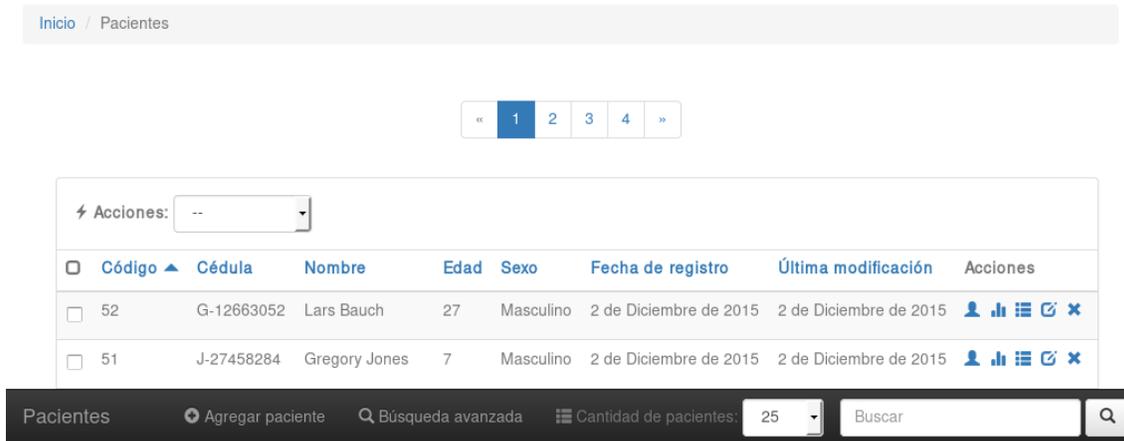
My actions

- house
Usuario
- Record: Jane Doe
(2016-05-18T03:55:22.038867+00:00) - RR - RR - Start:
2016-05-17T09:37:31-04:00
Canal
- Esteban Álvarez
(2015-07-29T04:08:17.834217+00:00) - Normal - test
Canal
- Esteban Álvarez
(2015-07-29T04:08:17.834217+00:00) - RR - RR
Canal
- Esteban Álvarez
(2015-07-29T04:08:17.834217+00:00) - 1.0 - 1.0
Punto
- Fibrilación auricular
Anomalia

Figura 5.3: Interfaz de administración

5.2.3. Iteración 3: Manejo de pacientes y datos aleatorios de prueba

Se creó la lista de pacientes que podemos ver en la figura 5.4, esta lista consiste en una tabla paginada con los principales datos de los pacientes; y con una lista de acciones que podemos aplicar sobre ellos, ya sea editarlos, visualizarlos o ver sus registros y diagnósticos. También podemos realizar búsquedas sencillas por nombre de paciente o cédula de identidad.



The screenshot shows a web interface for managing patients. At the top, there is a breadcrumb trail: "Inicio / Pacientes". Below it is a pagination control with buttons for "«", "1", "2", "3", "4", and "»", where "1" is highlighted. A dropdown menu labeled "Acciones:" is set to "--". The main content is a table with the following columns: "Código", "Cédula", "Nombre", "Edad", "Sexo", "Fecha de registro", "Última modificación", and "Acciones". Two rows of patient data are visible:

Código	Cédula	Nombre	Edad	Sexo	Fecha de registro	Última modificación	Acciones
52	G-12663052	Lars Bauch	27	Masculino	2 de Diciembre de 2015	2 de Diciembre de 2015	[Icons for actions]
51	J-27458284	Gregory Jones	7	Masculino	2 de Diciembre de 2015	2 de Diciembre de 2015	[Icons for actions]

At the bottom, there is a dark navigation bar with the text "Pacientes" and several icons. A search bar is present with the text "Búsqueda avanzada" and a "Buscar" button. A dropdown menu for "Cantidad de pacientes:" is set to "25".

Figura 5.4: Lista de pacientes

Para la creación y edición de pacientes se implementó un formulario común para ambos casos, se utilizó la funcionalidad interna de formularios que provee *Django*; que permite validar de manera sencilla y gestionar el guardado de los datos. Podemos ver un ejemplo de este formulario en la figura 5.5, en este caso la edición.

La interfaz para la ficha del paciente (figura 5.6) se hizo manteniendo la misma distribución del formulario, pero en vez de ingresar los datos se muestran como texto. La ficha del paciente está dividida en dos partes, las cuales podemos acceder por medio de pestañas en la interfaz. Aparte de los datos personales tenemos los datos clínicos (figura 5.7).

Inicio / Pacientes / Jane Doe / Editar

Datos personales Datos clínicos



Actualmente:
[patients/346a1a1d-f1e9-4fdd-b85d-e720a303024b.jpg](#)
 Limpiar

Modificar:

Nombre:	<input type="text" value="Jane"/>
Apellido:	<input type="text" value="Doe"/>
Cédula de identidad:	<input type="text" value="V"/> <input type="text" value="2344323"/>
Sexo:	<input type="text" value="Femenino"/>
Fecha de nacimiento:	<input type="text" value="08/03/1991"/>
Lugar de nacimiento:	<input type="text" value="Caracas"/>
Estado civil:	<input type="text" value="Soltero"/>
Nivel socioeconómico:	<input type="text" value="A"/>
Grado de instrucción:	<input type="text" value="Universitario"/>
Vivienda propia:	<input checked="" type="checkbox"/>
Dirección:	<input type="text" value="Caracas"/>

Figura 5.5: Edición de los datos del paciente

Inicio / Pacientes / Jane Doe

Datos personales Datos clínicos



Nombre:	Jane Doe
Cédula de identidad:	V-2344323
Sexo:	Femenino
Fecha de nacimiento:	8 de Marzo de 1991
Lugar de nacimiento:	Caracas
Edad:	26
Estado civil:	Soltero
Nivel socioeconómico:	A
Grado de instrucción:	Universitario
Vivienda propia:	Sí
Dirección:	Caracas

Figura 5.6: Ficha del paciente (datos personales)

Inicio / Pacientes / Jane Doe

Datos personales Datos clínicos



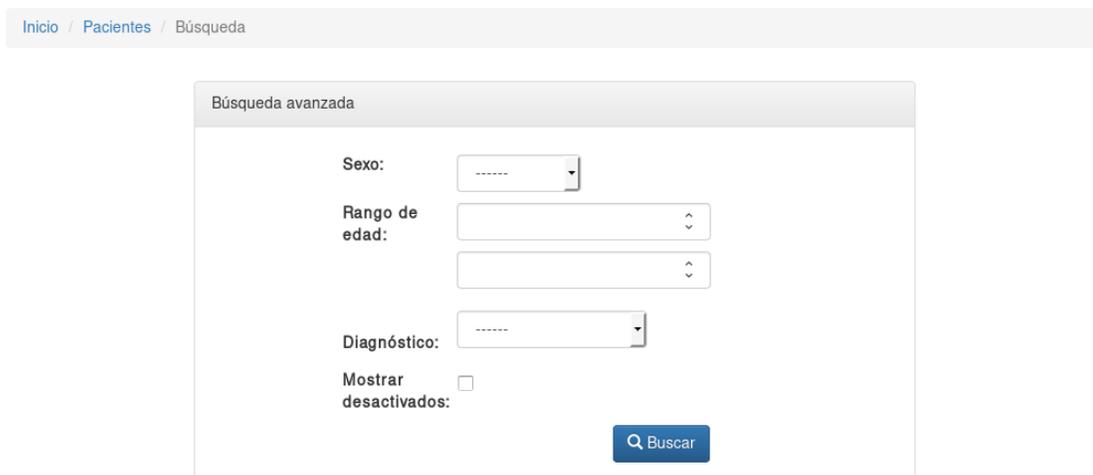
Fecha de ingreso:	24 de Noviembre de 2015 a las 06:37
Última modificación:	24 de Noviembre de 2015 a las 06:42
Ocupación:	Administrador
Antecedentes personales:	
<i>Hábitos personales:</i>	No
<i>Patológicos:</i>	No
Antecedentes familiares:	No
Examen físico:	N/A

Figura 5.7: Ficha del paciente (datos clínicos)

5.2.4. Iteración 4: Búsqueda y ordenamiento de los pacientes

Se implementan la función de ordenamiento y búsqueda. Para ordenar se hace por medio de enlaces en los títulos de las columnas como se puede observar nuevamente en la figura 5.4, ya que esta es la manera estándar de muchos sitios y aplicaciones web, permite no sólo ordenar la lista inicial, sino los resultados de una búsqueda.

La búsqueda avanzada se ejecuta por rango de edad, por sexo y diagnóstico, con la finalidad de poder agrupar los pacientes para realizar algún estudio manteniendo uno o más parámetros comunes, más adelante veremos que esto es muy útil a la hora de calcular estadísticas por grupo de pacientes. La interfaz del formulario de la búsqueda avanzada lo podemos ver en la figura 5.8.



The screenshot shows a web application interface with a breadcrumb trail at the top: "Inicio / Pacientes / Búsqueda". Below this is a form titled "Búsqueda avanzada". The form contains the following elements:

- Sexo:** A dropdown menu with a downward arrow.
- Rango de edad:** Two stacked input fields, each with a double-headed arrow on the right side, indicating a range.
- Diagnóstico:** A dropdown menu with a downward arrow.
- Mostrar desactivados:** A checkbox that is currently unchecked.
- Buscar:** A blue button with a magnifying glass icon and the text "Buscar".

Figura 5.8: Búsqueda avanzada

5.2.5. Iteración 5: Diagnósticos

Primero se realizó la lista de diagnósticos que podemos ver en la figura 5.9, acá tenemos un historial con todos los diagnósticos que se le han realizado al paciente, con sus respectivas observaciones y adjuntos (documentación para respaldar el diagnóstico, como exámenes de laboratorio, por ejemplo).

Los diagnósticos no se pueden borrar ni editar (salvo por un administrador), en cambio se puede agregar un nuevo diagnóstico si el actual no resulta ser acertado; para ello tenemos un formulario (figura 5.10) donde se especifica la anomalía, la observación y adjunto.

Inicio / Pacientes / Jane Doe / Diagnóstico

El diagnóstico del paciente Jane Doe ha sido guardado exitosamente

Historial de diagnósticos ➕ Agregar diagnóstico

Fecha	Realizado por	Diagnósticos	Descripción	
26 de Marzo de 2017	Gregory House	<input type="text" value="Fibrilacion auricular"/>		
15 de Marzo de 2017	Gregory House	<input type="text" value="Fibrilacion auricular"/>	Diagnóstico de prueba	

« 1 »

Figura 5.9: Lista de diagnósticos

Physiocardexp UCV

Salir

Inicio / Pacientes / Jane Doe / Diagnóstico / Nuevo diagnóstico

Nuevo diagnóstico

Diagnóstico:

Descripción:

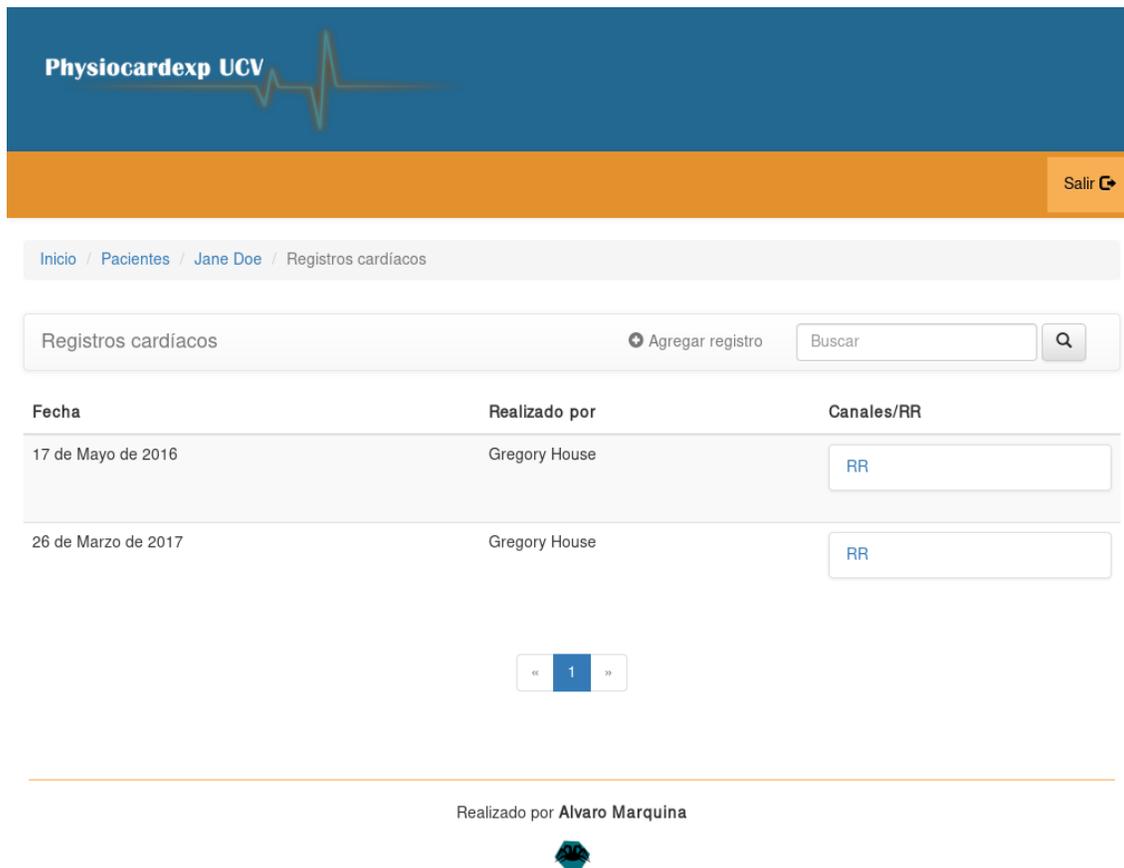
Adjunto: No file selected.

Realizado por Alvaro Marquina

Figura 5.10: Nuevo diagnóstico

5.2.6. Iteración 6: Lectura de datos para el electrocardiograma

Como se especificó en el capítulo anterior, la lectura se hizo por medio de un *script*, el cual lee el archivo especificado, lo interpreta y lo carga en la base de datos. Luego dicho registro aparece en la lista de la figura 5.11, una vez cargados los datos podemos graficarlo y calcular sus parámetros estadísticos.



The screenshot shows the Physiocardexp UCV web application interface. At the top, there is a blue header with the logo and an orange bar with a 'Salir' button. Below the header is a breadcrumb trail: 'Inicio / Pacientes / Jane Doe / Registros cardíacos'. A search bar contains 'Registros cardíacos' and a search icon. Below the search bar is a table with two rows of cardiac records. The table has three columns: 'Fecha', 'Realizado por', and 'Canales/RR'. The first row shows a record from '17 de Mayo de 2016' by 'Gregory House' with 'RR' channels. The second row shows a record from '26 de Marzo de 2017' by 'Gregory House' with 'RR' channels. Below the table is a pagination control showing '1' in a blue box. At the bottom, it says 'Realizado por Alvaro Marquina' with a small icon.

Fecha	Realizado por	Canales/RR
17 de Mayo de 2016	Gregory House	RR
26 de Marzo de 2017	Gregory House	RR

Figura 5.11: Registros cardíacos

5.2.7. Iteración 7: Lista de registros cardíacos

En la figura 5.11 podemos ver la lista resultante de esta iteración, en ella podemos ver datos básicos como la duración, hora de inicio y hora final. Cada registro tiene un conjunto de canales, estos representan cada derivación del ECG, o como en nuestro caso, la serie temporal RR sobre las que trabajan los especialistas, para resumir los canales son cada conjunto de puntos relacionados al registro.

5.2.8. Iteración 8: Instalación y configuración de las bibliotecas gráficas

Se instaló y se verificó el correcto funcionamiento de la biblioteca *Matplotlib* necesaria para generar los gráficos de los registros cardíacos y sus estadísticas. En las pruebas realizadas se generaron distintos gráficos con datos conocidos para cerciorarse del correcto funcionamiento. Una vez comprobado que el uso de la biblioteca era factible, se terminó de integrar a la aplicación.

5.2.9. Iteración 9: Generación de estimadores estadísticos

Se creó un formulario (figura 5.12), con la finalidad de tomar los parámetros requeridos por el especialista para la generación de los gráficos y estadísticas. En este formulario se piden datos como: el intervalo de tiempo a graficar, el tamaño en escala temporal de los segmentos en los cuales se divide el intervalo de tiempo a graficar (a partir de cada segmento temporal se podrá evaluar un estimador que representará un punto en la gráfica), y en el caso del histograma, el parámetro *bins*.

Paciente:	Jane Doe
Agregado:	17 de Mayo de 2016 a las 23:55
Tipo de registro:	RR
Tasa de muestreo:	500 muestras por segundo
Duración:	5:17:39.48
Hora de inicio:	17 de Mayo de 2016 a las 09:37
Hora de finalización:	17 de Mayo de 2016 a las 14:55
Valores sobre el total del registro	
STD (SDNN):	168,87 (s)
Media:	483,51 (s)

Indicadores	
Inicio	17/05/2016 9:37
Fin	17/05/2016 10:37
Tamaño del segmento	1 <input type="text"/> Minutos <input type="text"/>
Tipo de gráfica	Original <input type="text"/>
Generar	

Figura 5.12: Formulario de estadísticas del registro cardíaco/Serie temporal RR

Como se puede apreciar en la figura 5.12 se elige que tipo de gráfica se desea generar, o si en cambio se desea crear un reporte en PDF, también contiene una opción para mostrar todas las gráficas del dominio en el tiempo, como por ejemplo: la media y la desviación estándar.

En resumen, en esta vista se pueden ver, por ejemplo, estadísticas generales correspondiente a la serie temporal de los intervalos RR del registro cardíaco seleccionado para el estudio.

Luego del formulario creamos las distintas gráficas, donde cada una de las estadísticas se va generando con métodos distintos que reciben los mismos parámetros (a excepción del histograma que recibe *bins*), por ejemplo, en esta interacción, como se muestra en la columna a la izquierda de la figura 5.12, sólo se desarrollan de manera cuantitativa los estimadores básicos: media y desviación estándar.

En el caso de las gráficas del dominio del tiempo, se toma el intervalo y se divide en segmentos del tamaño en escala temporal indicado por el usuario por medio del formulario. Por cada segmento se calcula el valor estadístico del parámetro indicado (media, pnn50, por ejemplo), el resultado sería un punto (o vértice de la curva) en la gráfica como se puede observar en la figura 5.13.

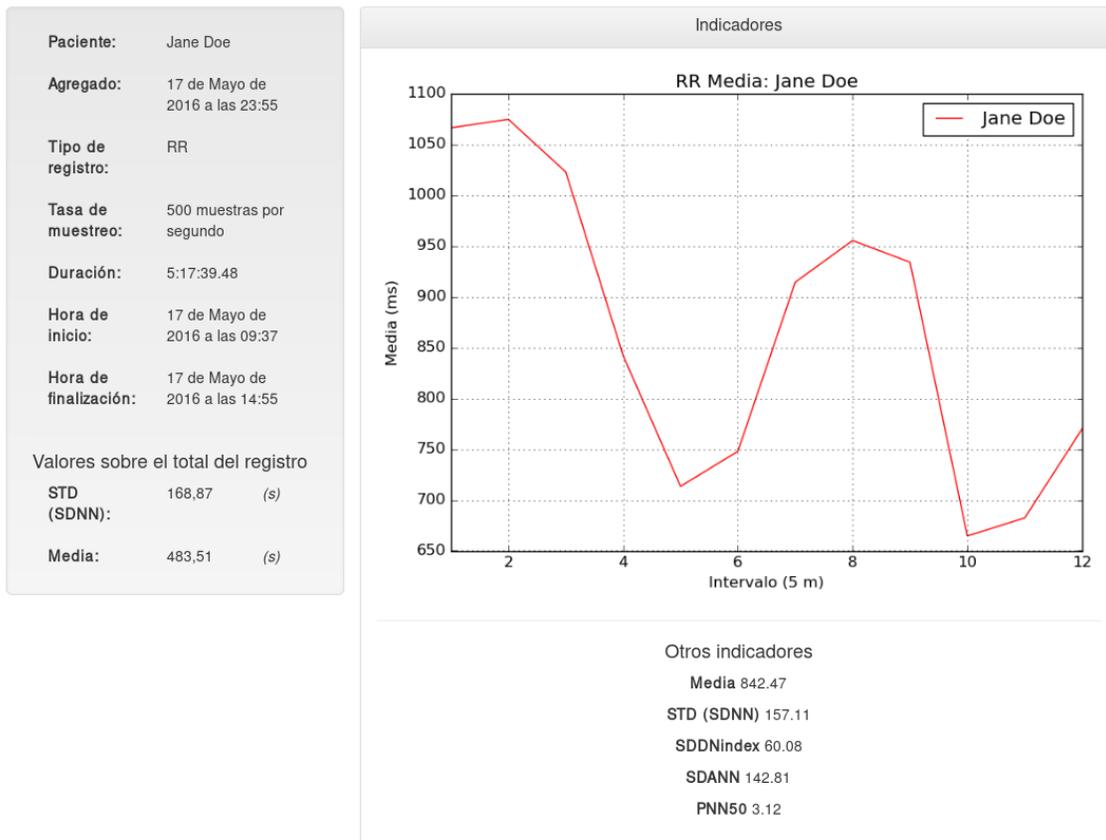


Figura 5.13: Media de los segmentos de la serie temporal RR

5.2.10. Iteración 10: Más estimadores estadísticos

Calculamos algunos de los estimadores/indicadores estadísticos que se usan en cardiología: SDNNindex, SDANN, pNN50 y rMSSD. Se anexan a las opciones disponibles de gráficas y a las estadísticas generales, según corresponda. Para iniciar el cálculo de un

estimador estadístico se indican como parámetros: el intervalo y sus segmentos, tal como se mencionó previamente, correspondiendo un valor o punto a cada segmento.

En la figura 5.14 podemos apreciar todas las gráficas resultantes de este paso junto a las de la iteración anterior. En este caso se eligió la opción “todas las gráficas” del formulario. Y en la figura 5.15 podemos apreciar un histograma como ejemplo de una salida gráfica con un único estimador estadístico.

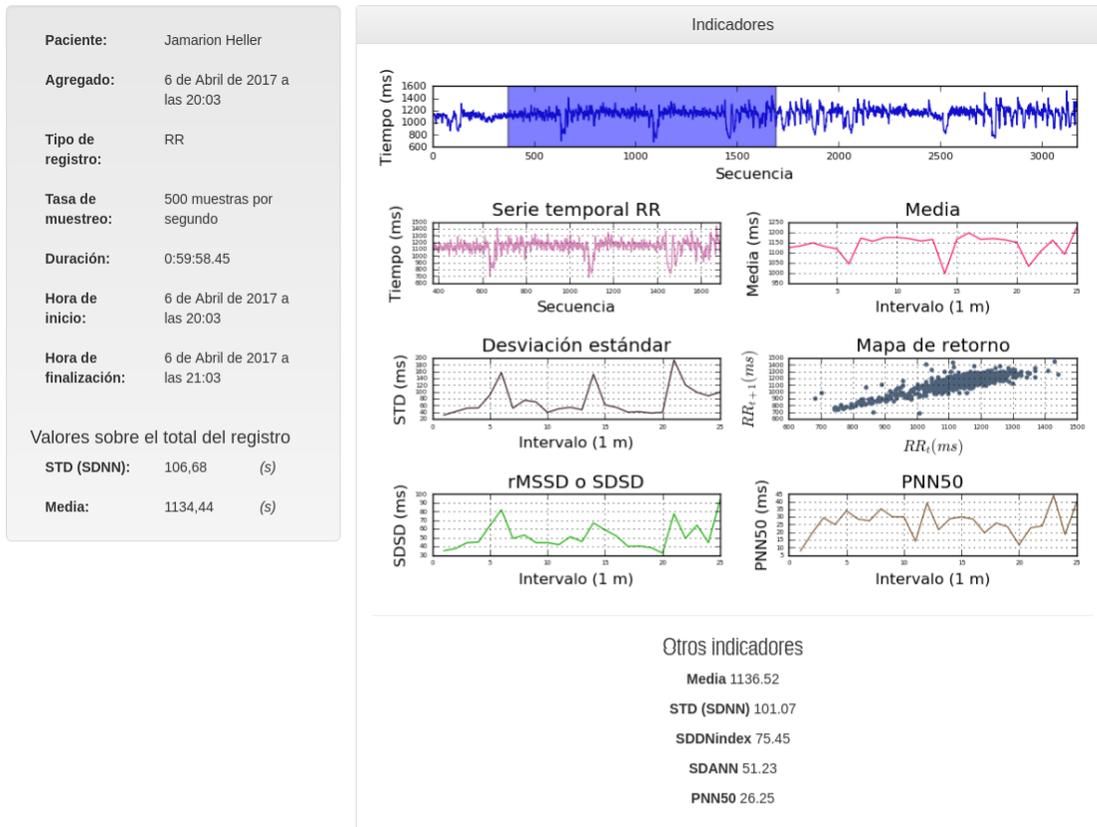


Figura 5.14: Todas las gráficas del dominio del tiempo

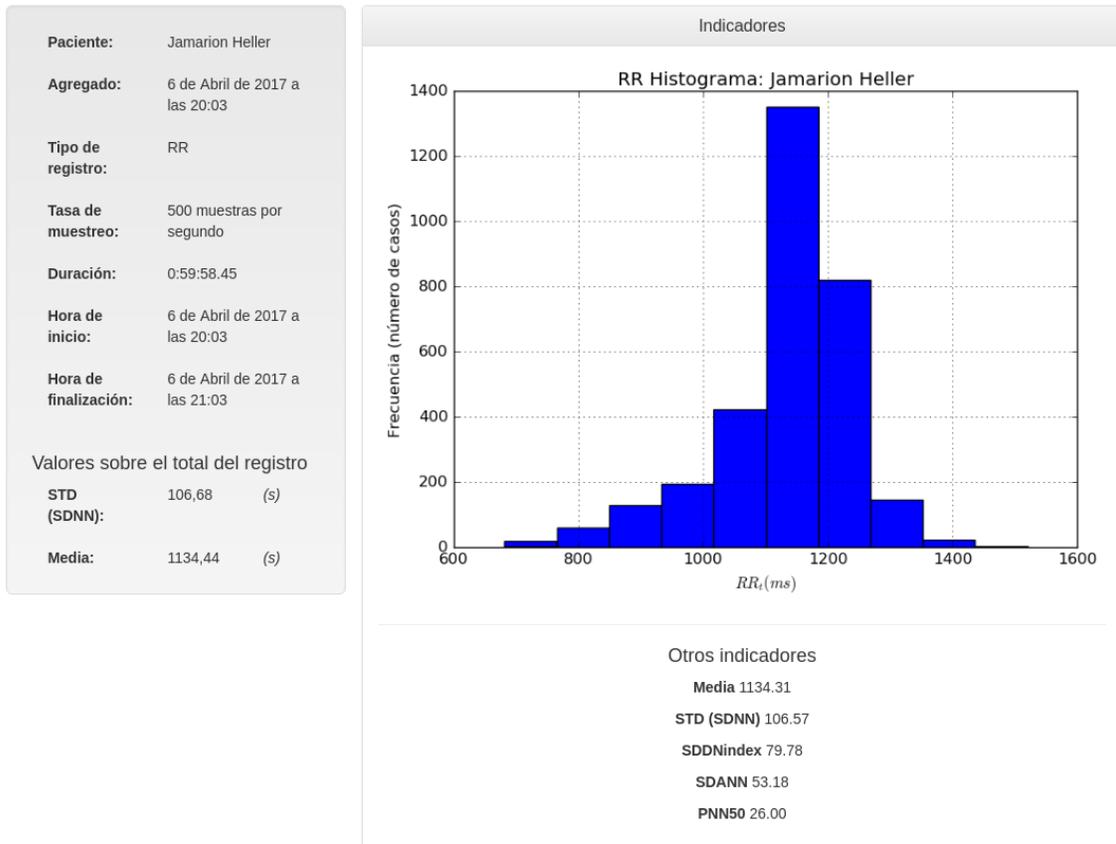


Figura 5.15: Histograma

5.2.11. Iteración 11: Estimadores frecuenciales

Los estimadores frecuenciales se calculan por medio de la transformada de *Fourier*, se separan las frecuencias en altas y bajas, y se procede a graficar, el resultado lo podemos apreciar en la figura 5.16. Se generan tres gráficas, en la primera, las altas (HF) y baja frecuencias (LF), en la segunda la potencia y en la tercera la relación entre altas y bajas frecuencias.

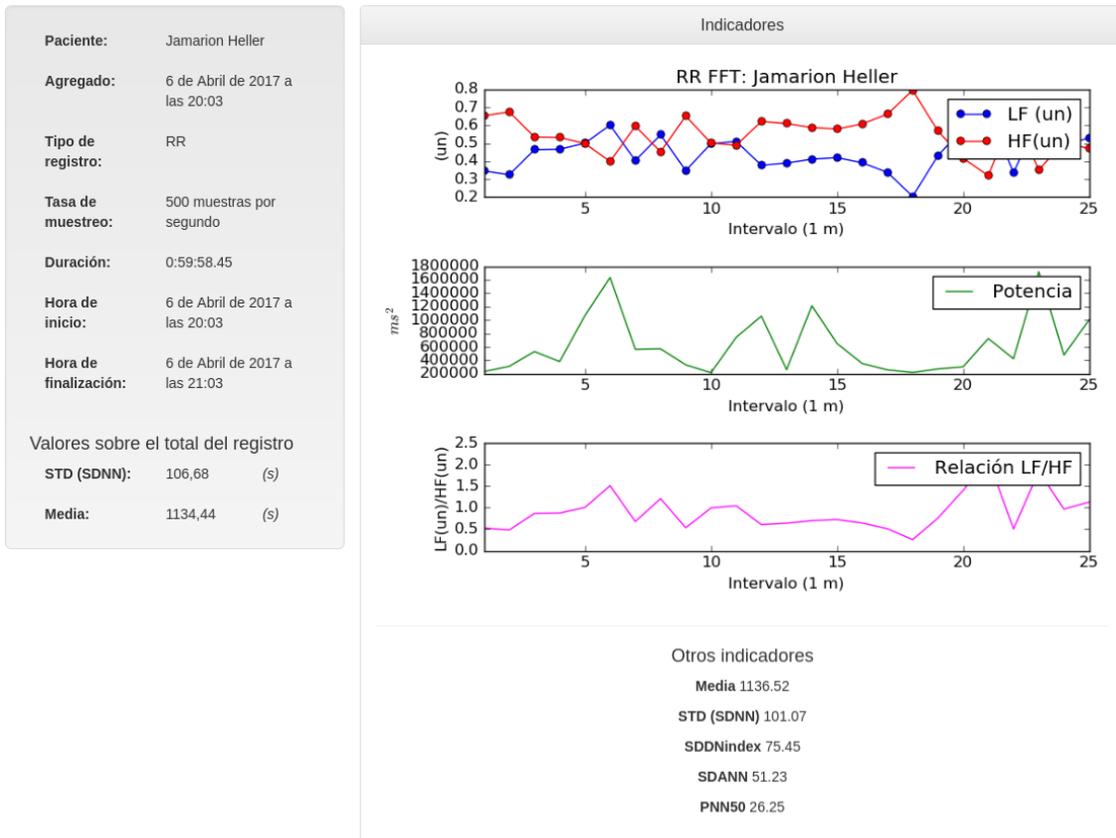


Figura 5.16: Estimadores de altas y bajas frecuencias para registros de un paciente

5.2.12. Iteración 12: Múltiples pacientes

Para calcular los estimadores con varios pacientes se seleccionan en la lista pacientes y en el desplegable de acciones se seleccionan “estadísticas”; luego se procede a rellenar el formulario que podemos ver en la figura 5.17. Este formulario es similar al de un solo paciente, con la diferencia que no genera un reporte en PDF. Un ejemplo de como lucen las gráficas lo podemos apreciar en la figura 5.18. Esta funcionalidad ofrece una comparación directa de perfiles, que muy bien se pueden corresponder a un mismo paciente o a distintos pacientes, con el fin de revisar el comportamiento de estimadores estadísticos en intervalos de períodos de tiempo seleccionados por el usuario, esta herramienta visual es de gran interés en el momento de hacer investigación por los especialistas en el área.



The image shows a web form titled "Parámetros" for generating graphs for multiple patients. On the left, there is a list of patient names: "Jane Doe" and "Esteban Álvarez". The main form contains the following fields:

- Inicio:** 09:39
- Fin:** 10:40
- Tamaño del segmento:** 1 (with a dropdown menu set to "Minutos")
- Indicador:** Media

A red "Generar" button is located at the bottom right of the form.

Figura 5.17: Formulario para las gráficas con múltiples pacientes

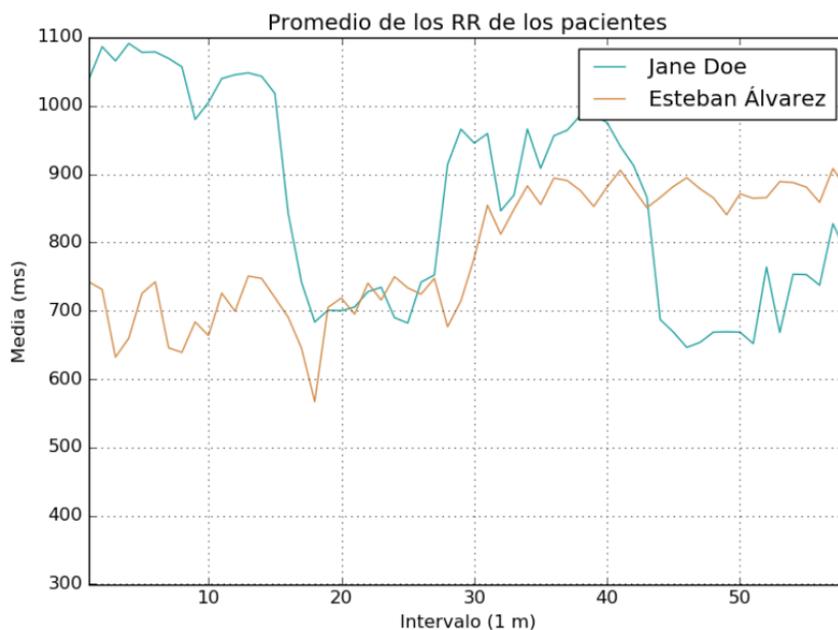


Figura 5.18: Media calculada para dos pacientes

5.2.13. Iteración 13: Reporte

Por último tenemos la salida final del sistema, el reporte en formato PDF, para generarlo se utiliza el formulario de estadísticas de usuario y recibe todos los parámetros, ya que genera todas las gráficas disponibles.

Como podemos apreciar en las figuras 5.19, 5.20 y 5.21 el reporte contiene los datos del usuario, el médico tratante, las estadísticas generales, los parámetros bajos los cuales fue generado, y un cuadro donde el especialista pueda anotar cualquier observación adicional una vez impreso el reporte.

Este reporte fue realizado bajo las especificaciones del personal de cardiología, y permite tener un soporte físico bajo los que realizar estudios y diagnósticos. Gracias al formato fijo de PDF está listo para su impresión.

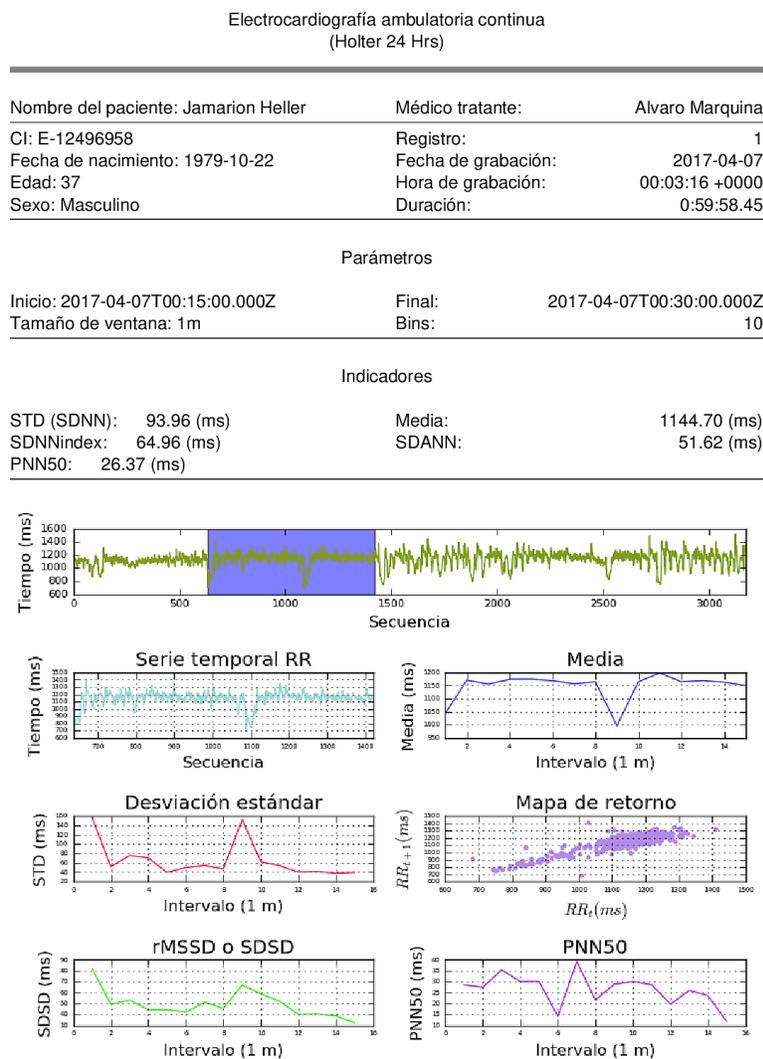


Figura 5.19: Reporte en pdf del registro del paciente, página 1

Electrocardiografía ambulatoria continua
(Holter 24 Hrs)

Nombre del paciente: Jamarion Heller

Médico tratante:

Alvaro Marquina

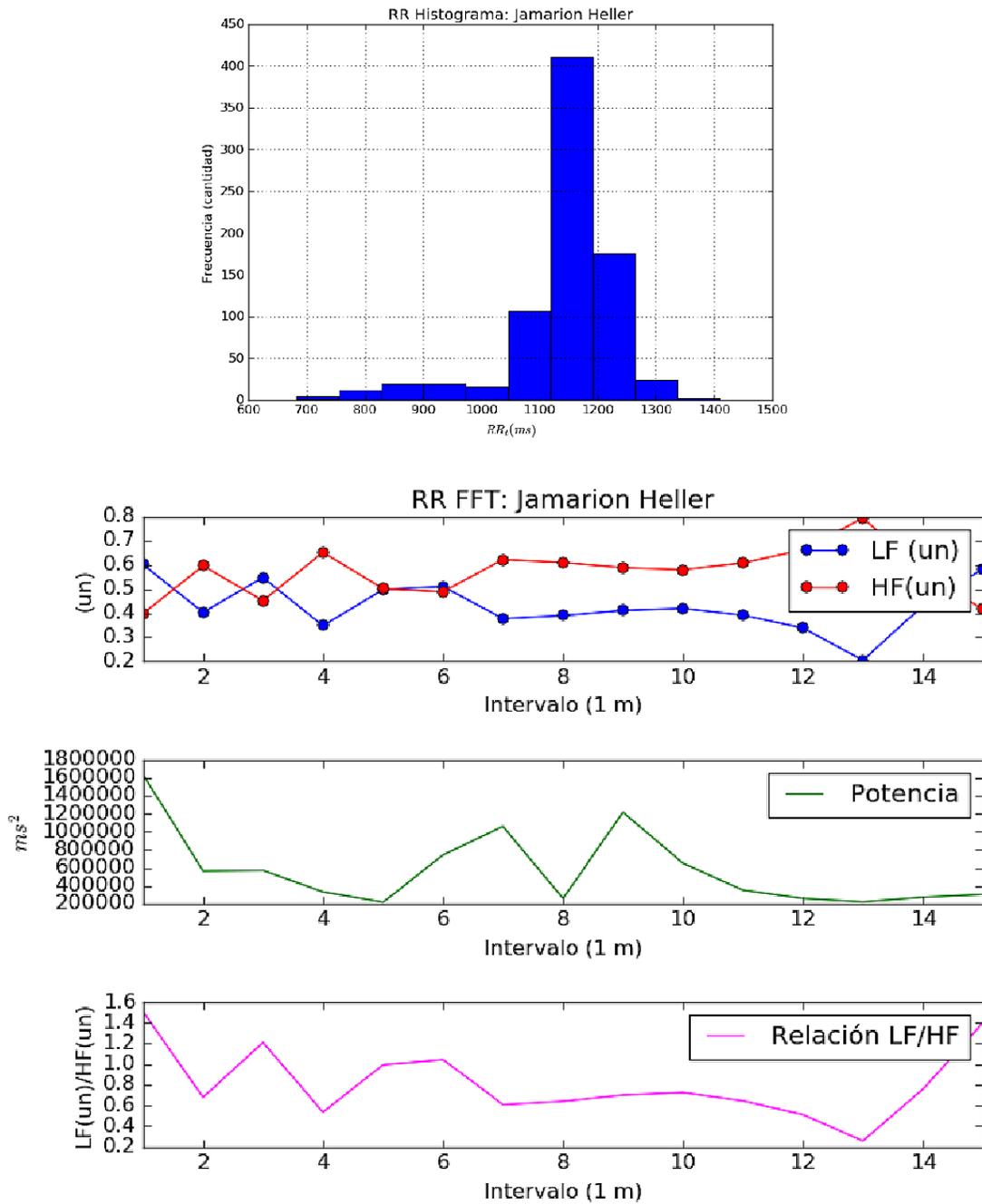


Figura 5.20: Reporte en pdf del registro del paciente, página 2

Electrocardiografía ambulatoria continua
(Holter 24 Hrs)

Nombre del paciente: Jamarion Heller

Médico tratante:

Alvaro Marquina

Observaciones

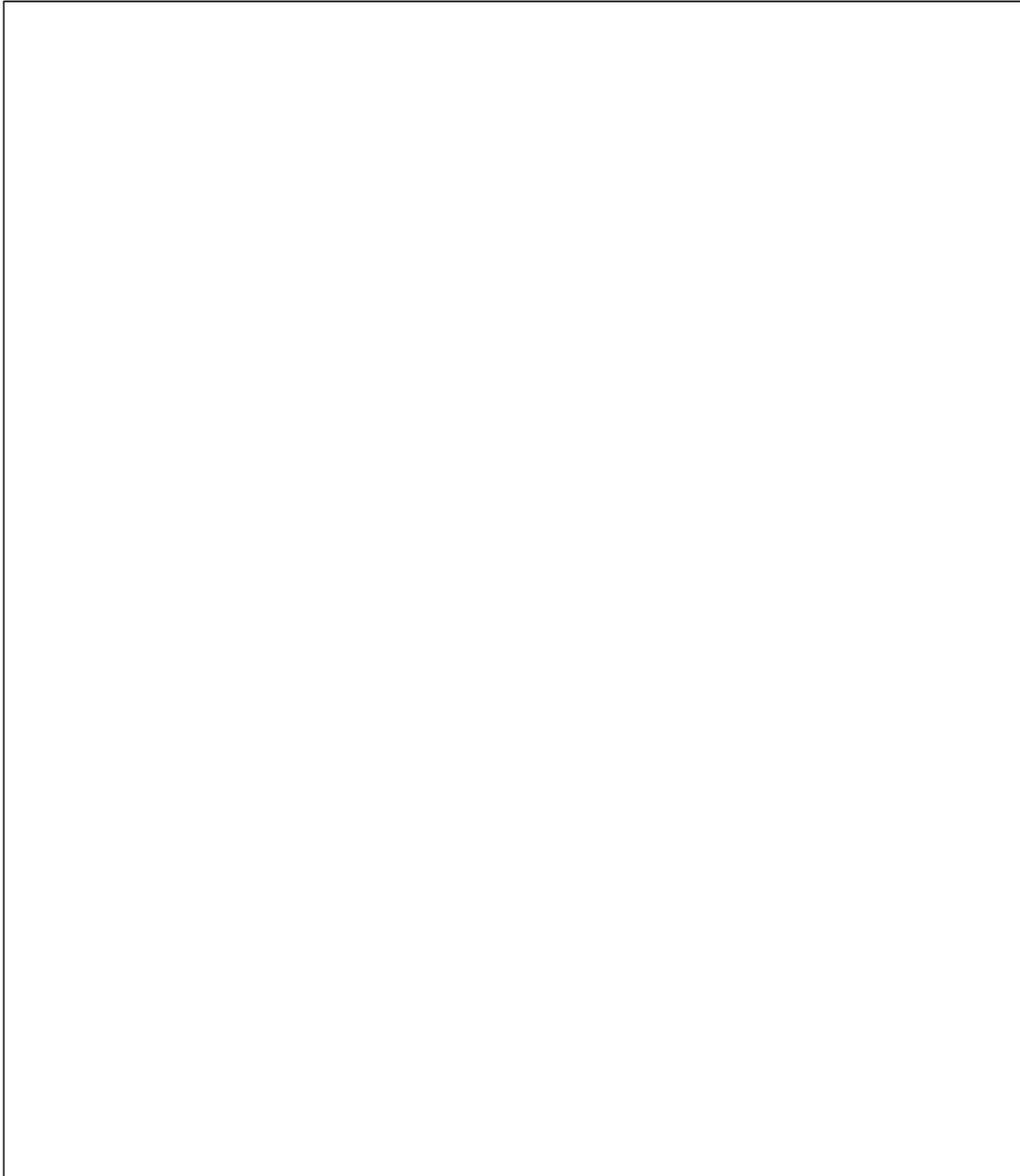


Figura 5.21: Reporte en pdf del registro del paciente, página 3

Capítulo 6

Conclusiones

Al inicio del proyecto se estableció un objetivo general que es desarrollar una base de datos de señales electrocardiográfica para su visualización y estudio. Para lograr este objetivo se siguieron una serie de pasos, lo cuales se establecieron desde el principio del proyecto como objetivos específicos, esta serie de pasos permitieron centrarse en el objetivo final y saber en qué punto del desarrollo se estaba situado.

Tomando en cuenta lo expuesto, podemos concluir lo siguiente:

- El estudio del corazón humano contiene muchas variantes que pueden estudiarse por medio de los indicadores tanto temporales como frecuenciales. Los especialistas utilizan estos datos para apoyar su diagnóstico. Este es un punto importante porque es el fuerte de nuestra aplicación.
 - El desarrollo de la aplicación de manera iterativa basándose en Scrum, es decir, un paso a la vez con la validación de los especialistas, permitió un desarrollo más fluido y eficaz.
 - La implementación fue hecha en el lenguaje de programación Python, respaldado por el *framework* Django, que permite una implementación rápida y organizada, gracias a la naturaleza del lenguaje y la gran cantidad de funcionalidades que provee tanto Python como Django, y su mayor bondad es que permite agregarle más funcionalidades futuras sin necesidad de modificar el código base. Por el lado del cliente se utilizó el *framework* Bootstrap, el cual permite construir interfaces sencillas en poco tiempo, gracias a esto también se tiene como resultado una interfaz escalable en el tiempo. Se logró con esto una aplicación que permite desarrollo futuro y fácilmente modificable en caso de requerirlo.
 - La aplicación resultante tiene varias características claves necesarias para el estudio y diagnóstico de anomalías cardíacas, que incluyen: la implementación de los indicadores estadísticos temporales y frecuenciales sobre la serie temporal de los intervalos RR, manejo de usuarios, manejo de pacientes y diagnósticos, y un reporte analítico
-

y gráfico en formato PDF, cumpliendo con las especificaciones y requerimientos de los especialistas.

- Todas las herramientas nombradas son código libre, al igual que la implementación, la mayor ventaja que nos provee esto es que tanto los datos como la aplicación se pueden utilizar en otros desarrollos. Esto permitirá que otras personas puedan colaborar a futuro en el proyecto, realizar aplicaciones que se conecten a la aplicación principal con la finalidad de usar sus datos, o herramientas derivadas de esta.
 - Finalmente se tiene un producto que cubre las necesidades del usuario, que permite su fácil utilización por parte del mismo, un mantenimiento relativamente sencillo para futuras funcionalidades, facilitando la utilización de los datos, pero sin perder su seguridad y cuidando la privacidad de los pacientes.
-

6.1. Recomendaciones

A propósito de que el presente trabajo se continúe desarrollando y ampliando sus capacidades, se enumeran a continuación algunas recomendaciones.

- Desarrollar una API (Interfaz de programación de interfaces, por sus siglas en inglés) que permita la utilización de los datos por una aplicación externa (ejemplo: Android, aplicación de escritorio, etc), manteniendo por supuesto las medidas de seguridad para mantener los datos seguros y lejos del alcance de usuarios no autorizados.
 - Generar reportes en lote y en segundo plano, para poder generar múltiples reportes por grupos de pacientes, y en segundo plano para mantener la fluidez de la aplicación.
 - Desarrollo de una aplicación móvil, que permita al especialista revisar los datos de sus pacientes desde su móvil Android o iOS, y así aumentar la portabilidad del sistema.
-

Bibliografía

- [1] Álvarez, E.; Jiménez, J.; Moleiro, F.; Rodríguez A. (2008). Designing an Alarm System for the Stratification of Risk of Cardiac Arrhythmias. IEEE Conference Proceeding Computer in Cardiology.
 - [2] Álvarez, E.; Jiménez, J.; Moleiro, F.; Rodríguez A. (2010). Characterizing atypical patterns of Heart Rate before Paroxysmal Ventricular Tachycardia. Medical Engineering & Physics 32, 1131–1136.
 - [3] Álvarez, Esteban (2010). Estudio de la predictibilidad en arritmias cardiacas. Tesis de Doctorado, Postgrado en Instrumentación. UCV. Publicación interna.
 - [4] Pérez, Aida (2015). Desarrollo de un Sistema Prototipo de Adquisición, Registro y Transmisión de Señales ECG para Pruebas de Holter. Trabajo de Grado de Maestría. Postgrado en Instrumentación. UCV. Publicación Interna.
 - [5] Colmenarez Castillo, Valentina (2016). Reconocimiento de Patrones de Registros de Electrocardiografía Dinámica de Holter 24 horas. Trabajo Especial de Grado. Escuela de Computación. Facultad de Ciencias, UCV. Publicación Interna.
 - [6] Ramírez, Oswaldo (2016). Análisis de Estimadores Estadísticos de Riesgo Cardíaco. Trabajo Especial de Grado. Escuela de Física. Facultad de Ciencias, UCV. Publicación Interna.
 - [7] National Geographic. (5 de septiembre del 2010). Corazón. España: National geographic. Recuperado de <http://www.nationalgeographic.es/ciencia/salud-y-cuerpo-humano/heart-article>
 - [8] Fundación española del corazón. (s.f.). Cómo funciona el corazón humano. España: Fundación española del corazón. Recuperado de <http://www.fundaciondelcorazon.com/informacion-para-pacientes/como-funciona-el-corazon.html>
 - [9] Espinosa, Carmen. (8 de mayo del 2016). ¿Cuáles son las capas que recubren el corazón?. About en español. Recuperado de <http://enfermedadescorazon.about.com/od/nociones-basicas/a/Capas-Del-Corazon-Pericardio-Endocardio-Y-Miocardio.htm>
-

-
- [10] Salud y bienestar. (s.f.). Capas del corazón. Recuperado de <http://lasaludi.info/capas-del-corazon.html>
- [11] Texas Heart Institute. (Agosto 2016). Las arterias coronarias. Texas, EEUU: Texas Heart Institute. Recuperado de http://www.texasheartinstitute.org/HIC/Anatomy_Esp/corooa_sp.cfm
- [12] Instituto Químico biológico. (s.f.). Cardiología – Potenciales de membrana. Recuperado de <http://www.iqb.es/cardio/fisiologia/fisio01.htm>
- [13] Arritmias Cardiacas. (29 de mayo del 2014, actualmente fuera de línea). La electricidad del corazón. Recuperado de https://web.archive.org/web/20140403082046/http://arritmiascardiacas.net/ac2/?page_id=22
- [14] Olmo, M., Nave, R. (s.f.). El Nodo Sinusal: El Marcapasos Natural del Cuerpo. Hyperphysics, Georgia State University. Recuperado de <http://hyperphysics.phy-astr.gsu.edu/hbasees/biology/sanode.html>
- [15] Olmo, M., Nave, R. (s.f.). Fenómenos Eléctricos en el Corazón. Hyperphysics, Georgia State University. Recuperado de <http://hyperphysics.phy-astr.gsu.edu/hbasees/biology/heartelec.html#c1>
- [16] Espinosa, Carmen. (5 de diciembre del 2015). ¿Cómo funciona el sistema eléctrico del corazón?. About en español. Recuperado de <http://enfermedadescorazon.about.com/od/El-corazon/a/El-Sistema-De-Conduccion-Del-Corazon.htm>
- [17] Rodas, Gil; Pedret Carballido, Carles; Ramos, Juan; Capdevila Lluís. (2008). Variabilidad de la frecuencia cardíaca: concepto, medidas y relación con aspectos clínicos. Barcelona, España: Sociedad Española de Medicina del Deporte. Recuperado de http://femede.es/documentos/Variabilidad_41_123.pdf
- [18] MedlinePlus. (2016). Monitor Holter (24 horas). Rockville Pike, Estados Unidos: U.S. Department of Health and Human Services. Recuperado de <https://medlineplus.gov/spanish/ency/article/003877.htm>
- [19] Saceda Corralo, David. (2017). Holter, qué es y en qué consiste. Webconsultas. Recuperado de <http://www.webconsultas.com/pruebas-medicas/holter-12058>
- [20] Electrocardiograma (ECG). (s.f.). Electrocardiograma (ECG). Bupa Salud. Recuperado de <https://www.bupasalud.com/salud-bienestar/vida-bupa/electrocardiograma-ecg>
- [21] Adobe.com. (2017). ¿Qué son las aplicaciones webs y las páginas web dinámicas?. Adobe. Recuperado de <https://helpx.adobe.com/es/dreamweaver/using/web-applications.html>
-

- [22] Álvarez, Miguel Ángel. (2014). Qué es MVC. Desarrolloweb. Recuperado de <https://desarrolloweb.com/articulos/que-es-mvc.html>
- [23] MDN (Mozilla Developer Network). (2016). HTML—MDN. Mozilla Developer Network. Recuperado de <https://developer.mozilla.org/es/docs/Web/HTML>
- [24] MDN (Mozilla Developer Network). (2016). CSS—MDN. Mozilla Developer Network. Recuperado de <https://developer.mozilla.org/es/docs/Web/CSS>
- [25] MDN (Mozilla Developer Network). (2017). JavaScript—MDN. Mozilla Developer Network. Recuperado de <https://developer.mozilla.org/es/docs/Web/JavaScript>
- [26] Bootstrap. (2017). Bootstrap – The world’s most popular mobile-first and responsive front-end framework. Bootstrap. Recuperado de <http://getbootstrap.com/>
- [27] W3Schools. (s.f.). AJAX Introduction. W3Schools. Recuperado de https://www.w3schools.com/xml/ajax_intro.asp
- [28] Rouse, Margareth. (s.f.). What is a database (DB)?. SearchSQLServer. Recuperado de <http://searchsqlserver.techtarget.com/definition/database>
- [29] HowStuffWorks. (s.f.). What are relational databases?. HowStuffworks. Recuperado de <http://computer.howstuffworks.com/question599.htm>
- [30] Python 2.7.13 Documentation. (s.f.). General Python FAQ. Python Docs. Recuperado de <https://docs.python.org/2/faq/general.html#what-is-python>
- [31] Django Project. (s.f.). The web framework for perfectionists with deadlines. Django Project. Recuperado de <https://www.djangoproject.com/>
- [32] ScrumAlliance. (2012). Scrum, una descripción. ScrumAlliance. Recuperado de <https://goo.gl/gyxmKV>
- [33] Manifesto for Agile Software Development. (2001). Manifesto for Agile Software Development. Recuperado de <http://agilemanifesto.org/>
- [34] Alaimo, Martín. (2013). Proyectos ágiles con Scrum. Buenos Aires, Argentina: Kleer. Recuperado de <http://media.kleer.la/kleer-proyecto-agiles-con-scrum.pdf>
- [35] Biomedical Issues. (5 de noviembre del 2010). Entendiendo la transformada discreta de Fourier. Recuperado de <http://biorigo.blogspot.com/2010/11/entendiendo-la-transformada-discreta-de.html>
- [36] Ramírez, Gustavo. (2002). Sistema para el desarrollo de protocolos de diagnóstico cardiológico (pp. 11–12). Escuela de Física, Facultad de Ciencias UCV, Única Edición.
-

- [37] Hernández, Eugenio. (2006). Matemática de las señales. España: Universidad Autónoma de Madrid. Recuperado de https://www.uam.es/personal_pdi/ciencias/ehernan/Otros/Senyalas-v3.pdf.
- [38] [Conceptos Básicos de Señales y Sistemas]. (s.f.). Recuperado de <http://chitita.uta.cl/cursos/2012-2/0000435/recursos/r-1.pdf>.
- [39] Guerrero Martínez, Juan F. (2010-2011). Procesado Digital de Bioseñales. Universidad de Valencia, Ingeniería biomédica. Recuperado de http://ocw.uv.es/ingenieria-y-arquitectura/1-5/ib_material/IB_T4_OCW.pdf.
- [40] Rojas, Kervin; Romero, Carmen; Romero, Pedro. (2013). Modelo de procesamiento digital de señales cardíacas desarrollado en Matlab. Barranquilla, Colombia: Universidad de la Costa (CUC). Recuperado de <http://publicaciones.urbe.edu/index.php/telematique/article/viewArticle/2523/html>.
- [41] Pisarello, María Inés; Picaza, Carlos Álvarez; Monzón, Jorge. (s.f.). Separación de Frecuencias No Deseadas en la Señal Cardíaca utilizando ICA. Corrientes, Argentina: Facultad de Ciencias Exactas, Universidad Nacional del Nordeste. Recuperado de <https://goo.gl/WttzA6>.
- [42] Consorcio ITACA. Investigación Traslacional de Arritmias Cardíacas Secundarias a Canalopatías. (s.f.). Canalopatías. Conceptos. Potencial de Acción Cardíaco. Recuperado de <https://www.itaca.edu.es/potencial-accion-cardiaco.htm>.
- [43] G. Arfken. (1981). Métodos matemáticos para físicos (pp. 739-770). Editorial Diana, 1ra. Edición.
-