



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

---

**FACULTAD DE INGENIERÍA**

**Conformación de una base  
de datos a partir del análisis  
de los desfibriladores  
automáticos implantables**

**INFORME DE ACTIVIDADES PROFESIONALES**

Que para obtener el título de

**Ingeniera en Sistemas Biomédicos**

**P R E S E N T A**

Andrea Silvana Salinas Durán

**ASESOR DE INFORME**

M.I. Efraín Albor Ramírez



**Ciudad Universitaria, Cd. Mx., 2024**

## Índice

|           |   |           |
|-----------|---|-----------|
| <b>1</b>  | <b>Conformación de una base de datos a partir del análisis de los desfibriladores automáticos implantables.</b> | <b>3</b>  |
| <b>2</b>  | <b>Introducción</b>   | <b>3</b>  |
| 2.1       | Objetivo  | 4         |
| <b>3</b>  | <b>Descripción de la empresa o medio en que labora</b>  | <b>4</b>  |
| 3.1       | <i>Boston Scientific</i>  | 4         |
| 3.2       | Descripción del puesto de trabajo o labor que desempeñó   | 4         |
| <b>4</b>  | <b>Antecedentes</b>   | <b>5</b>  |
| 4.1       | Definiciones clínicas   | 5         |
| 4.2       | Automatización con <i>Robotic Process Automation</i>  | 6         |
| 4.3       | OCR   | 6         |
| 4.4       | Python  | 7         |
| 4.5       | Tableau   | 7         |
| <b>5</b>  | <b>Definición del problema o contexto de la participación profesional</b>                                       | <b>8</b>  |
| <b>6</b>  | <b>Metodología utilizada</b>  | <b>8</b>  |
| 6.1       | Extracción de gráficas y valores por cada sensor  | 10        |
| 6.2       | Procesamiento de señales y extracción de datos  | 13        |
| 6.3       | Identificación de datos   | 18        |
| 6.4       | Creación de <i>Dashboard</i> para la visualización de datos.  | 20        |
| <b>7</b>  | <b>Resultados</b>   | <b>25</b> |
| 7.1       | Extracción de gráficas y valores por cada sensor  | 25        |
| 7.2       | Procesamiento de señales y extracción de datos  | 27        |
| 7.3       | Identificación de los datos   | 33        |
| 7.4       | Dashboard para la visualización de datos  | 35        |
| 7.4.1     | Primera parte del Dashboard Heart Logic   | 35        |
| 7.4.2     | Segunda parte del <i>Dashboard Heart Logic</i>  | 37        |
| <b>8</b>  | <b>Conclusiones</b>   | <b>39</b> |
| <b>9</b>  | <b>Bibliografía</b>   | <b>40</b> |
| <b>10</b> | <b>Anexos</b>   | <b>41</b> |
| 10.1      | Partes del código: Métodos de Binarización  | 41        |

|        |   |    |
|--------|---|----|
| 10.2   | Partes del código: Operaciones morfológicas ..... | 42 |
| 10.2.1 | Erosión.....                                      | 42 |
| 10.2.2 | Dilatación.....                                   | 42 |
| 10.2.3 | Apertura .....                                    | 42 |
| 10.2.4 | Close .....                                       | 43 |

## **1 Conformación de una base de datos a partir del análisis de los desfibriladores automáticos implantables.**

### **2 Introducción**

La insuficiencia cardíaca (IC) se caracteriza por la incapacidad del corazón para bombear suficiente cantidad de sangre rica en oxígeno a lo largo de todo el organismo. Esta cardiopatía se desarrolla progresivamente a medida que la función de bombeo se debilita o cuando se enfrenta a dificultades para llenar adecuadamente el órgano con sangre entre latidos, afectando potencialmente ambos lados del corazón [1].

Cuando la insuficiencia cardíaca impacta el lado izquierdo del corazón, la capacidad para bombear sangre oxigenada al resto del cuerpo se ve comprometida; en el caso del lado derecho, la capacidad para bombear suficiente sangre a los pulmones, donde se realiza la recolección de oxígeno, se ve afectada [1].

Los pacientes con insuficiencia cardíaca experimentan hospitalizaciones frecuentes, lo cual está asociado a una elevada tasa de mortalidad, reingresos y una carga económica significativa [2]. A pesar de las recomendaciones para el monitoreo del peso y los síntomas como medida de control de la IC [3], no se ha demostrado que estas prácticas reduzcan las hospitalizaciones [4]. Dada la importancia clínica y económica de las hospitalizaciones, se ha desarrollado un método eficaz para detectar el empeoramiento de la IC.

"Heart Logic", un desfibrilador implantable, utiliza múltiples sensores fisiológicos de alta sensibilidad para diagnosticar y monitorizar de forma remota y personalizada la insuficiencia cardíaca. Este dispositivo ha sido validado para proporcionar detección temprana de signos de empeoramiento de la cardiopatía con semanas de antelación [5].

El índice compuesto Heart Logic, considerado el dato más relevante, indica si un paciente padece insuficiencia cardíaca, debiendo ser menor a 16 para considerar al paciente como estable en esta condición. Este índice se calcula mediante un algoritmo que utiliza datos recopilados por los sensores, como el primer y tercer ruido cardíaco (S1 y S3), vinculados a la contracción ventricular; la impedancia torácica, asociada a la acumulación de líquido y edema pulmonar; la frecuencia respiratoria, relacionada con patrones de respiración rápida y superficial que indican dificultad para respirar; y la frecuencia cardíaca, que refleja la actividad y estado general del paciente [6].

Aunque la información de los sensores y el índice compuesto se utiliza instantáneamente durante la visita, solo se recopilan gráficos a lo largo de seis meses. Sin embargo, estas representaciones gráficas son densas y dificultan la lectura diaria. Además, los valores proporcionados por los sensores en el informe son solo del día de la visita de seguimiento y descarga.

La obtención y procesamiento de datos de las gráficas mediante lenguajes de programación permitirá obtener información clave en diferentes intervalos de tiempo, facilitando al médico la predicción de posibles insuficiencias cardíacas o mejoras. Esta recopilación de datos también puede ser crucial para detectar nuevos síntomas de trastornos cardiovasculares y anticipar las necesidades de los pacientes que requieren atención especializada. La creación de una base de datos proporciona un registro continuo de cada usuario, contribuyendo a mejorar la calidad de atención al paciente a lo largo del tiempo.

## 2.1 Objetivo

El objetivo de este trabajo es presentar un reporte del proyecto que se elaboró en la empresa, el objetivo del proyecto es proporcionar mayor información a los doctores que atienden pacientes con insuficiencia cardíaca que tienen un desfibrilador implantable “*Heart Logic*”, por medio de la conformación y visualización de una base de datos que se extrae del funcionamiento de los sensores que proporcionan información que muestran con antelación necesidades del paciente.

## 3 Descripción de la empresa en que labora

### 3.1 *Boston Scientific*

Misión: *Boston Scientific* se dedica a transformar vidas a través de soluciones médicas innovadoras que mejoran la salud de los pacientes en todo el mundo. Nuestro trabajo se rige por valores fundamentales que definen la cultura de *Boston Scientific* y empoderan a nuestros empleados [7].

Cuidado • Innovación significativa • Alto rendimiento • Colaboración global • Diversidad • Espíritu ganador

Cuidado: Actuamos con integridad y compasión para apoyar a los pacientes, clientes, nuestras comunidades y entre nosotros.

Diversidad: Aceptamos la diversidad y valoramos los talentos, ideas y experiencias únicas de nuestros empleados.

Colaboración Global: Trabajamos en colaboración para buscar oportunidades globales que extiendan el alcance de nuestras soluciones médicas.

Alto Rendimiento: Nos esforzamos por lograr un alto rendimiento para beneficiar a nuestros pacientes, médicos y accionistas.

Innovación Significativa: Fomentamos un ambiente de creatividad para transformar nuevas ideas en servicios innovadores y soluciones que crean valor para pacientes, clientes y empleados.

Espíritu Ganador: Nos adaptamos al cambio y actuamos con rapidez, agilidad y responsabilidad para mejorar aún más la atención al paciente.[7]

### 3.2 Descripción del puesto de trabajo o labor que desempeñó

#### **Becaria de *Supply Chain Transformation***

- Brindar soporte a la Transformación de la Cadena de Suministro, en una amplia variedad de iniciativas enfocadas en proyectos de Automatización, Visibilidad de Datos y Evitación de Costos.

- Trabajar en objetivos o proyectos de áreas funcionales específicas que ~~pueden requerir~~ requieren de investigación, trabajo en equipo, desarrollo de presentaciones, diseño de estrategias y mantenerse al tanto de las tendencias del mercado o los productos de la industria.

- Actividades a realizadas:

- a. Procedimientos estándar de la Cadena de Suministro relacionados con Atención al Cliente, Logística, Flete, Inventario de Campo.
- b. Base de datos, aplicación móvil y gestión y soporte web.
- c. Uso de SAP, incluidas transacciones, informes y simulación.
- d. Análisis de datos (*Tableau Desktop, Tableau Prep, PowerBI, Python, R*).
- e. Políticas y procedimientos.
- f. Evaluación del inventario gestionado por el proveedor.
- g. Transformación digital dentro y fuera de la cadena de suministro.
- h. Gestión de relaciones con proveedores.

## 4 Antecedentes

### 4.1 Definiciones clínicas

#### Frecuencia respiratoria

Se refiere a la cantidad de respiraciones que genera una persona por minuto. Se calcula en reposo y se cuantifican las respiraciones o elevación del tórax por minuto. El promedio normal de la frecuencia respiratoria es entre 12 y 20. [8]

#### Frecuencia cardiaca

Es el conteo de latidos por minuto. El corazón actúa bombeando sangre hacia todos los órganos, pero para un correcto funcionamiento se requiere que lo haga a una determinada presión y frecuencia. La frecuencia cardiaca en reposo normal para los adultos es entre 60 y 100 latidos por minuto. [9]

#### Impedancia Torácica

La impedancia intratorácica es la resistencia eléctrica entre un electrodo implantado en el ventrículo derecho y un desfibrilador colocado en la región subclavicular. Si aumenta el líquido torácico la impedancia será menor, debido a que la corriente viaja con menor resistencia a través del líquido. Tiene una relación inversamente proporcional, por lo que si los pulmones se congestionan la impedancia disminuye. [10]

#### Desfibrilador automático implantable

El DAI por sus siglas Desfibrilador Automático Implantable. Se encarga de restablecer el ritmo cardiaco normal, su funcionamiento es a través de descargas eléctricas en caso de la aparición de arritmias. El DAI puede estimular el corazón al igual que un marcapasos si el ritmo cardiaco es muy lento. [11]

#### Primer y tercer ruido cardiaco

El corazón provoca unas vibraciones llamados ruidos o tonos cardiacos, estos se pueden detectar a través de la auscultación cardiaca, la cual es una técnica para valorar el funcionamiento del corazón y sus válvulas. Se lleva a cabo a través del fonendoscopio.

Para captar los ruidos se recorre la región precordial y se centran en 4 áreas (aórtico, pulmonar, tricúspide y mitral). En la auscultación normal se captan los dos primeros ruidos.

El tercer y cuarto ruido se producen por una menor distensibilidad y, por consiguiente, mayor rigidez en el ventrículo. [12]

En el primer ruido ocurre el cierre de las válvulas aurícula-ventriculares (mitral y tricúspide) al final de la diástole. Coincide con la contracción isovolumétrica ventricular. Su intensidad depende de la velocidad de cierre y estado de las válvulas auriculo-ventriculares.[12]

El tercer ruido es el inicio de la diástole en la fase de llenado ventricular rápido. Vibración de la pared ventricular porque es más rígida por dilatación ventricular debida a sobrecarga de volumen. Se puede oír en insuficiencia cardíaca, insuficiencia mitral, pericarditis constrictiva, anemia o en situaciones fisiológicas como en niños o embarazo.[12]

#### 4.2 Automatización con *Robotic Process Automation*

La Automatización Robótica de Procesos (RPA, por sus siglas en inglés) implica la creación de robots, comúnmente conocidos como bots, que ejecutan tareas específicas mediante una serie de instrucciones programadas por un individuo a través de un software. Esta tecnología tiene como objetivo replicar y realizar de manera eficiente actividades repetitivas, contribuyendo así a la optimización del tiempo y la eficacia de los procesos.

Dentro del panorama de software utilizado para implementar la RPA, se destacan varias opciones, entre las cuales se encuentran Automation Anywhere, Blue Prism, UI Automation, Work Fusion y UiPath [13].

En el marco de este proyecto, se optó por el uso de Automation Anywhere como el software principal para la creación de dos RPA's o bots. Las funciones disponibles en Automation Anywhere se distinguen por su intuitividad, permitiendo a los usuarios diseñar procesos mediante la disposición visual de bloques de acciones en la pantalla principal. Estas acciones abarcan una variedad de funciones, como la apertura de archivos, la extracción mediante OCR, la captura de pantalla, la creación de tablas, la escritura, entre otras. Una vez que se ha completado el diseño del proceso, es posible ejecutar el código y verificar su funcionamiento.

Es relevante destacar que Automation Anywhere proporciona una interfaz amigable que facilita la implementación de automatizaciones, incluso para aquellos usuarios que no poseen una amplia experiencia en programación. Además, la versatilidad de las acciones disponibles en este software permite abordar una gama diversa de tareas, desde las más simples hasta las más complejas, en diferentes contextos y sectores industriales.

#### 4.3 OCR

La obtención de datos clínicos se llevó a cabo mediante el empleo de la automatización, específicamente mediante la Robotic Process Automation (RPA), utilizando la función Optical Character Recognition (OCR), o Reconocimiento Óptico de Caracteres en inglés. Esta tecnología tiene la capacidad de identificar imágenes, símbolos o caracteres en diversos idiomas y digitalizarlos de manera rápida y eficiente [14].

En cuanto al funcionamiento del OCR, esta acción se encuentra disponible en la plataforma Automation Anywhere y se caracteriza por su sencillez de uso. Consiste en arrastrar el bloque de acción al proceso correspondiente y configurar los parámetros necesarios, que incluyen el nombre de la ventana desde la cual se extraerán los datos, la captura de la región de esa ventana, la selección del idioma de los datos y la asignación de una variable para almacenar el resultado.

En un análisis más detallado del funcionamiento específico del OCR, se pueden identificar las siguientes etapas:

1. Adquisición de imagen: Los documentos deseados se leen y convierten en datos binarios. La función de OCR clasifica las áreas claras como fondo y las áreas oscuras como texto mediante un proceso de umbralización.
2. Procesamiento previo de limpieza: El OCR realiza una limpieza de la imagen, eliminando ruidos y manchas digitales, o suavizando los bordes de las imágenes.
3. Reconocimiento de texto: Se utilizan dos tipos de algoritmos para el reconocimiento de texto:
  - A. Coincidencia de patrones: Se aísla una imagen de carácter llamada glifo y se compara con un glifo almacenado de manera similar. El reconocimiento se logra si el glifo almacenado tiene una fuente y una escala similares a las de la entrada.
  - B. Extracción de características: Los glifos se dividen o descomponen en características como líneas, circuitos cerrados, dirección de línea e intersecciones de línea. Luego, se utilizan estas características para encontrar la mejor coincidencia o el vecino más cercano entre los glifos almacenados.
4. Procesamiento posterior de texto: Finalmente, el sistema transforma los datos de texto extraídos en un archivo computarizado [14].

#### 4.4 Python

Para la extracción y el procesamiento de señales se empleó un lenguaje de programación informática denominado Python, el cual está compuesto por diversas bibliotecas especializadas. Dado que el propósito consistía en adquirir datos provenientes de una señal representada en formato de imagen (png), se hizo uso principalmente de la biblioteca OpenCV, reconocida por su aplicación en el procesamiento de imágenes y vídeo.

Adicionalmente, se integraron otras bibliotecas en el código, entre las que se destaca NumPy, contribuyendo al cálculo numérico y análisis de datos. Matplotlib desempeñó un papel fundamental en la generación de gráficos en dos dimensiones para verificar la efectividad en la extracción de datos. Asimismo, se incorporó la biblioteca Pandas con el propósito de exportar los datos extraídos de la gráfica a un documento en formato Excel.

#### 4.5 Tableau

Tableau se presenta como una herramienta especializada en el análisis de datos y en la inteligencia empresarial. Esta plataforma analítica facilita la visualización de datos provenientes de diversos tipos de sensores en intervalos temporales específicos, desempeñando un papel crucial en la satisfacción de las necesidades del paciente mediante una estratégica utilización de dicha información.



## **5 Definición del problema o contexto de la participación profesional**

En la actualidad, se encuentra disponible en el mercado un desfibrilador automático implantable conocido como "Heart Logic", diseñado para su implementación en pacientes que padecen insuficiencia cardíaca. Este dispositivo se vale de un algoritmo que posibilita la monitorización de diversos parámetros, tales como la frecuencia cardíaca nocturna, los movimientos respiratorios, la impedancia torácica, la actividad física y la intensidad de los tonos cardiacos, con la finalidad de anticipar eventos clínicos significativos.

Los individuos que adquieren este dispositivo deben someterse a revisiones periódicas cada seis meses, durante las cuales se descarga un archivo en formato PDF. Este documento presenta valores recopilados de los sensores, abarcando los componentes mencionados anteriormente en el transcurso de un día específico (el día de la descarga). A su vez, proporciona gráficas que abarcan periodos de seis meses para cada componente o sensor. Dada la extensión de los periodos contemplados en estas representaciones gráficas, se pierde precisión y valiosa información en cada revisión, especialmente en aspectos críticos como la frecuencia cardiaca y respiratoria, la impedancia torácica, la frecuencia respiratoria, el índice de insuficiencia cardíaca y la variabilidad de la frecuencia cardíaca. No obstante, el médico especialista solo cuenta con la información del día de descarga y los gráficos imprecisos, sin tener una visualización y recopilación concreta de los datos.

La extracción de valores durante el procesamiento de cada gráfica permite obtener información más precisa en intervalos de tiempo más reducidos, adaptados a las necesidades de análisis de los especialistas médicos. Al contar con una base de datos ya procesada, se convierte en una herramienta de visualización más precisa que proporciona, mediante gráficas mensuales, una visión detallada de los cinco sensores. Esto posibilita la detección temprana de irregularidades y la predicción de eventos clínicos no deseados.

El propósito de este proyecto es desarrollar una base de datos que incluya la información más representativa del paciente a lo largo del tiempo, registrando cualquier irregularidad en cada componente. Esta información se presentará mediante un Dashboard, facilitando su visualización tanto para los médicos como para los pacientes. La representación más precisa de la información permitirá realizar un análisis predictivo más efectivo ante cualquier indicio que requiera atención especializada en enfermedades cardíacas.

## **6 Metodología utilizada**

La metodología empleada para crear la base y la visualización de datos de los pacientes con desfibriladores automáticos implantables se presenta en la [ilustración 1]. En esta metodología, se registraron dos pacientes que asistieron a sus visitas semestrales en fechas distintas, lo cual generó dos archivos en formato pdf. Cada archivo contiene información de los sensores y gráficas correspondientes.

A partir de estos documentos, se identificaron los datos a extraer, como se muestra en la [Ilustración 1]. En total, se extrajeron dieciséis datos. De estos, diez corresponden a los valores de los sensores registrados en el día de descarga. Estos datos son de vital importancia para el seguimiento y control del funcionamiento del desfibrilador automático implantable.

Además de la información capturada por los sensores, se extrajeron otros seis datos que componen la información personal del paciente. Estos elementos son relevantes para la identificación y caracterización de cada paciente en la base de datos. Entre los detalles personales extraídos se encuentran la fecha de nacimiento, el número de serie del dispositivo, el género, entre otros.

Una vez que se identificaron los valores en el archivo PDF, se implementó un Proceso Robótico Automatizado (RPA), haciendo uso de la función de Reconocimiento Óptico de Caracteres (OCR) para interpretar el texto contenido en el documento [Ilustración 1]. Con la finalización del RPA, se logró extraer los dieciséis valores a un archivo CSV.

Las gráficas cruciales para este proyecto abarcan los ruidos cardíacos S1 y S3, la frecuencia respiratoria, la impedancia torácica y el índice Heart Logic [Ilustración 1]. Para la obtención de estas gráficas, se desarrolló otro RPA mediante diversas funciones, permitiendo la extracción de las gráficas en formato de imagen y su organización en carpetas respectivas.

Con la adquisición de las imágenes, se implementaron tres códigos en Python. En el presente trabajo, se detallará de manera exhaustiva la creación de tres códigos distintos, principalmente debido a sus diferencias de entrada. El primero se enfoca en el procesamiento de la gráfica del "índice", el segundo código se encarga del procesamiento de las demás gráficas, y el tercer código se destina a la identificación de datos, como se presenta en la [Ilustración 1].

Mediante el archivo de Excel generado por Python y el archivo CSV obtenido a través del RPA, se procedió a cargarlos en Tableau. Se crearon distintas hojas con características específicas, como alertas, promedios, demografía, entre otras. En última instancia, se consolidaron estas hojas en un panel de control ("Dashboard") para su visualización conjunta.

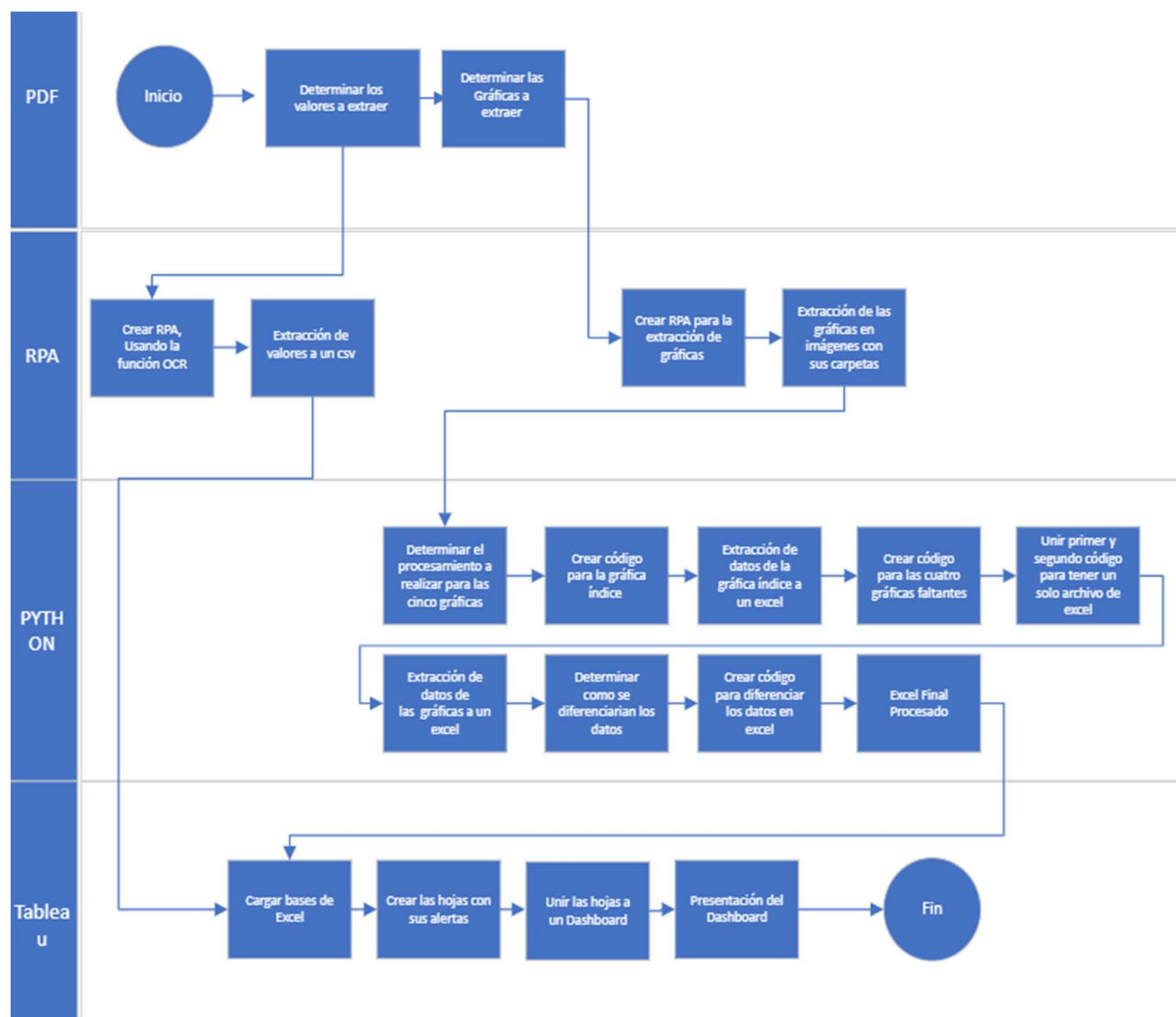
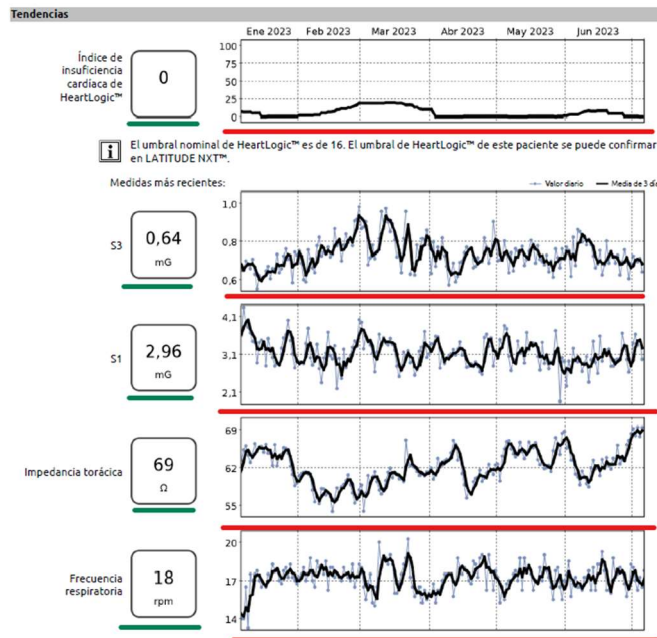


Ilustración 1. Diagrama de metodología

### 6.1 Extracción de gráficas y valores por cada sensor

La fase inicial consistió en la implementación de un Proceso Robótico Automatizado (RPA) con el propósito de extraer los valores correspondientes a cada sensor, así como información relevante del paciente.

El archivo generado por el desfibrilador adopta un formato en PDF que consta de dos páginas. Los datos provenientes de los sensores, cuya extracción se buscaba realizar, se encontraban alojados en recuadros ubicados a un costado de sus respectivas representaciones gráficas, como se observa en la [Ilustración 2].

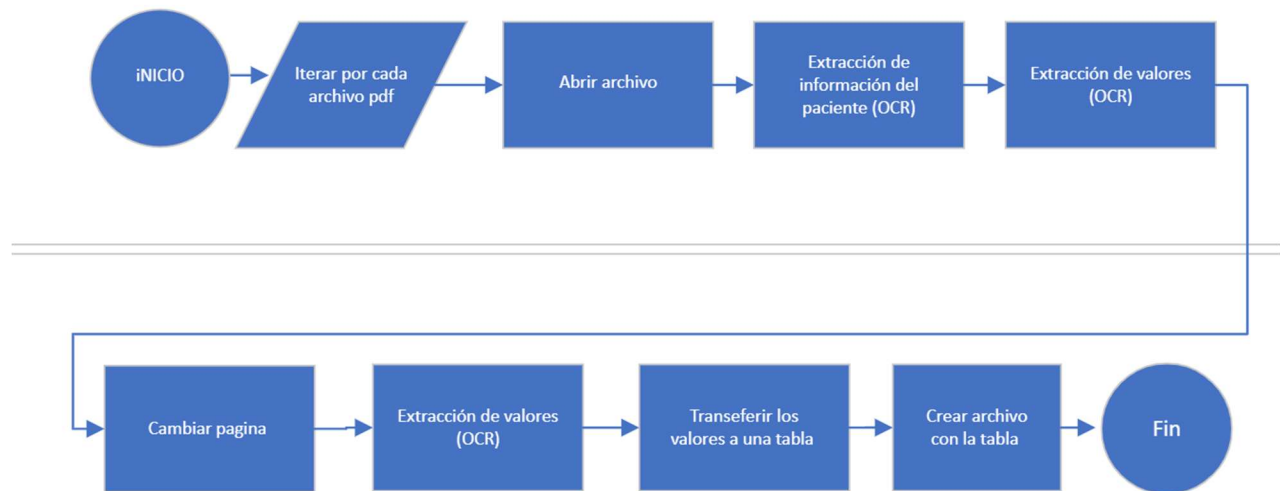


*Ilustración 2. Valores que extraer mediante OCR de cada sensor. De color rojo se observan las gráficas de cada sensor y de color verde los valores que las acompañan.*

Se implementó un reconocimiento de caracteres mediante la acción de Optical Character Recognition (OCR) en el Proceso Robótico Automatizado (RPA). Cada valor de interés y la información relacionada al paciente fueron identificados por el OCR y almacenados en variables distintas. El código revela múltiples acciones asociadas a cada valor reconocido, evidenciadas en la [Ilustración 3]. Es importante señalar que después de cada acción de reconocimiento, se introdujo un retardo de 100 [ms], un componente crucial en el proceso para asegurar que las acciones no se ejecuten secuencialmente sin esperar la finalización de la acción precedente.

La fase conclusiva del procedimiento incorporó una serie de acciones destinadas a transferir las variables a una tabla, la cual constituiría la primera base de datos en formato Excel.

Este proceso fue aplicado a los dos archivos en formato PDF, correspondientes a dos pacientes diferentes, dando como resultado un único archivo de Excel derivado de la extracción. Se diseñó el RPA considerando futuros análisis de más pacientes, implementando una acción denominada "Loop" que abarca todo el conjunto de acciones del código. Esta acción establece un bucle repetitivo para el número de archivos disponibles y concluye tras extraer los datos del último documento. Se destaca por la presencia de un paralelogramo en el proceso, que representa el bucle implementado desde el inicio de la operación [Ilustración 3].



*Ilustración 3. Diagrama del RPA creado para la extracción de valores por cada sensor.*

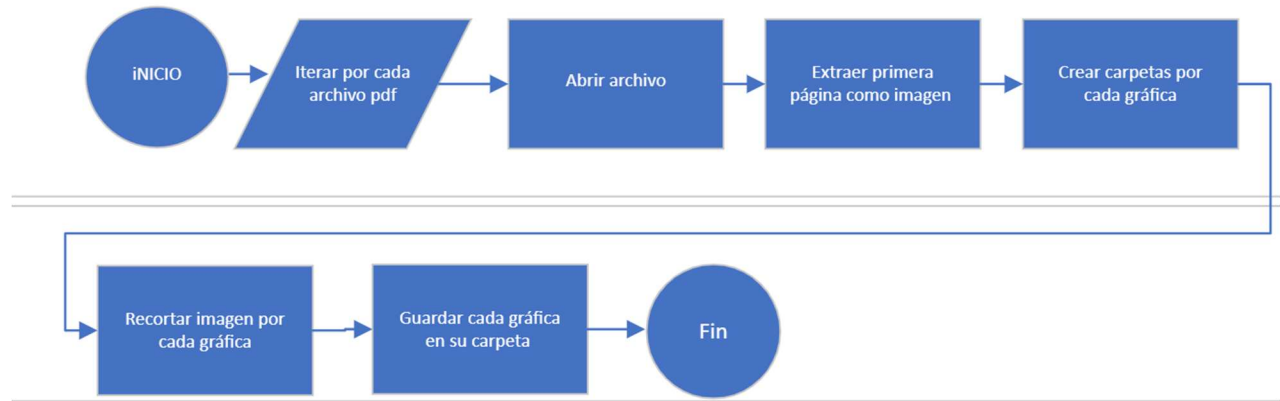
Para la obtención de representaciones gráficas, se implementó un segundo Proceso de Automatización Robótica (RPA). Durante su desarrollo, se adoptó un enfoque distinto, en virtud de que la funcionalidad OCR en Automation Anywhere únicamente reconoce caracteres o texto, excluyendo la interpretación de imágenes.

El procedimiento se centró en la conversión de los archivos PDF de interés a formato de imagen. Se llevó a cabo la conversión exclusivamente de la primera página de cada archivo PDF, considerando que en dicha página se exhiben las gráficas correspondientes a los cinco sensores sujetos a extracción [ilustración 4].

Subsecuentemente, se procedió a la edición de cada imagen mediante el recorte preciso de la gráfica asociada a cada sensor, definiendo sus coordenadas de origen  $(x_0, y_0)$  y las dimensiones a recortar  $(w, h)$ . Posteriormente, se generó una carpeta para cada categoría de gráfica y serie del archivo, designando el nombre de la ventana del archivo como parte del nombre de la carpeta, seguido del título de la gráfica, por ejemplo, ("HF\_06.pdf - Adobe Acrobat Reader (32-bit) \_FrecResp"). La creación de estas carpetas fue instrumental para la correcta operación del componente IQ BOT Pre-processor, dada su restricción de requerimientos que únicamente contempla el almacenamiento en directorios.

Se exhibe el bucle mediante la representación gráfica de un paralelogramo desde el inicio del proceso, con la finalidad de ilustrar la iteración inherente al mismo en cada archivo (paciente) [Ilustración 4].

En última instancia, cada imagen fue almacenada en la correspondiente carpeta, clasificada por tipo de gráfica y nombre de la ventana del archivo PDF (paciente).



*Ilustración 4. Proceso para la extracción de gráficos de los cinco sensores.*

## 6.2 Procesamiento de señales y extracción de datos

Para la manipulación de señales, se optó por el empleo del lenguaje de programación Python junto con cuatro de sus bibliotecas, a saber, OpenCV, NumPy, Matplotlib y Pandas. El siguiente análisis detalla el procedimiento implementado:

En una fase inicial, se procedió a la identificación de similitudes y disparidades entre las gráficas, con el propósito de obtener una comprensión sobre el procesamiento que podría ser aplicado para su depuración. Al identificar las gráficas desiguales, únicamente en un tipo (Índice Heart Logic) por archivo (en un total de dos gráficas), se observó la presencia de líneas negras punteadas formando rectángulos en su fondo. Este fenómeno presentaba un inconveniente al extraer los datos de la gráfica del sensor Índice Heart Logic, dado que los datos de interés estaban enmarcados por una línea negra de grosor considerable y no lineal. En el proceso de identificación, el programa no reconocía el área de interés debido al color negro compartido por ambas estructuras. Ante esta dificultad, se desarrolló el código siguiendo los siguientes pasos [Ilustración 5]:

1. Recortar la imagen para quitarle el borde negro a la gráfica, para esto se usó el tamaño de la imagen y se le resto los pixeles a recortar.
2. Convertir la imagen a escalas de grises para simplificar algunos cálculos y operaciones posteriores. El procesamiento.
3. Aplicar filtro que incluye realizar una binarización de la imagen, es decir, convierte la imagen en binaria donde los pixeles se clasifican como blanco o negro según su umbral. El umbral mínimo utilizado fue de "0" y el valor máximo que se asignara a los pixeles en blanco después de la binarización fue "255".

Los dos métodos de binarización que se utilizaron fueron binarización invertida y método de umbralización automática Otsu.

En la binarización invertida los píxeles con valores por encima del umbral se establecen en negro, mientras que los píxeles por debajo del umbral se establecen en blanco.

El método de umbralización automática Otsu determina un umbral óptimo basado en el histograma de intensidad de la imagen. Después de aplicar este filtro, la imagen "binaria" contendrá líneas punteadas representadas por píxeles blancos y el fondo negro.

Para obtener una comprensión más detallada sobre "binarización" y su aplicación en el procesamiento de imágenes, se sugiere consultar el anexo adjunto.

4. Aplicar la operación morfológica de open para eliminar las líneas punteadas, ajustando el *kernel*. La operación morfológica OPEN es una combinación de dos operaciones básicas: erosión seguida de dilatación. La idea detrás de esta operación es eliminar pequeños objetos y ruido de fondo mientras se preservan los objetos más grandes y estructuras importantes en la imagen. Para obtener una comprensión más detallada sobre "erosión y dilatación" y su aplicación en el procesamiento de imágenes, se sugiere consultar el anexo adjunto.

Matemáticamente, se puede expresar como:

$$A \circ B = (A \ominus B) \oplus B \quad \text{_____}(1)$$

Donde:

$A$  representa la imagen original.

$B$  es el elemento estructurante o kernel utilizado para la dilatación y la erosión.

$\oplus$  denota la operación de dilatación.

$\ominus$  denota la operación de erosión.

La secuencia de operaciones en la operación morfológica OPEN es importante. Primero, se aplica la erosión  $A \ominus B$ , que tiene el efecto de reducir el tamaño de los objetos en la imagen. Luego, se aplica la dilatación  $(A \ominus B) \oplus B$ , lo que revierte parcialmente el efecto de la erosión al expandir los objetos, pero aún conserva el efecto de eliminación del ruido y pequeños detalles [20 y 21]

En resumen, la operación morfológica OPEN es útil para suavizar y eliminar el ruido de fondo mientras se conservan las características más grandes y significativas de los objetos en una imagen.

5. Aplicar la operación morfológica de cierre a la imagen, para rellenar los huecos generados. La operación morfológica de cierre, denotada como  $A \cdot B$ , es la combinación de dos operaciones morfológicas básicas: dilatación seguida de erosión. En términos matemáticos, la operación de cierre se define como:

$$A \cdot B = (A \oplus B) \ominus B \quad \text{_____}(1)$$

La secuencia de operaciones en la operación de cierre es inversa a la de la operación de apertura. Primero, se aplica la dilatación  $A \oplus B$ , que tiene el efecto de expandir los objetos en la imagen. Luego, se aplica la erosión  $(A \oplus B) \ominus B$ , lo que revierte parcialmente el efecto de la dilatación al

contraer los objetos, pero aún conserva el efecto de rellenar pequeños huecos o discontinuidades en los objetos.

La operación de cierre es útil para cerrar pequeños huecos en los objetos y para suavizar los contornos de los objetos en una imagen. Es especialmente útil en la eliminación de pequeños agujeros o puntos de ruido en objetos.

6. Encontrar contornos en la imagen cerrada.
7. Rellenar de blanco los contornos encontrados y obtener la imagen invertida, se le colocó el nombre de "imagen invertida" debido a que después del relleno se tuvieron que cambiar los colores.
8. Obtener las dimensiones de la imagen invertida.  
Primero se definen los parámetros "alto\_img" y "ancho\_img", cuyos valores representan la altura y el ancho de la imagen original "imagen invertida", respectivamente. Estas dimensiones se obtienen utilizando el método shape de NumPy, que devuelve una tupla que contiene la altura, el ancho y el número de canales de la imagen.
9. Obtener las dimensiones del rectángulo.  
Una vez determinado el tamaño de la imagen, se definen las dimensiones del rectángulo. "alto\_rect" y "ancho\_rect", los cuales representan la altura y el ancho de los rectángulos en los que se dividirá la imagen.
  - "alto\_rect" se define como igual a la altura de la imagen original.
  - "ancho\_rect" se define como la división entera del ancho de la imagen original entre la cantidad de muestras ("samples"). Esto significa que el ancho de cada rectángulo será una fracción del ancho total de la imagen, determinada por el número de muestras (240).
10. Procesamiento para obtener datos de la imagen invertida.
  - Se itera sobre la imagen invertida en dos bucles "for" anidados para dividirla en rectángulos de igual tamaño.
  - Primer bucle "for" (Iteración vertical)  
Este bucle itera sobre las coordenadas verticales de la imagen. El parámetro "range" especifica que la iteración comienza desde 0 y termina en "alto\_img" (altura de la imagen), con un paso dado por "alto\_rect" (la altura del rectángulo). Esto significa que la imagen se divide verticalmente en rectángulos de altura "alto\_rect".
  - Segundo bucle "for" (Iteración horizontal)  
Este bucle itera sobre las coordenadas horizontales de la imagen. Al igual que el bucle anterior, el parámetro range especifica que la iteración comienza desde 0 y termina en ancho\_img (el ancho de la imagen), con un paso dado por ancho\_rect (el ancho del rectángulo). Esto significa que cada rectángulo vertical se divide horizontalmente en segmentos de ancho ancho\_rect.  
Entonces, el rango con el que se itera en ambos bucles está determinado por el tamaño de la imagen (alto\_img y ancho\_img) y el tamaño del rectángulo (alto\_rect y ancho\_rect). Esto

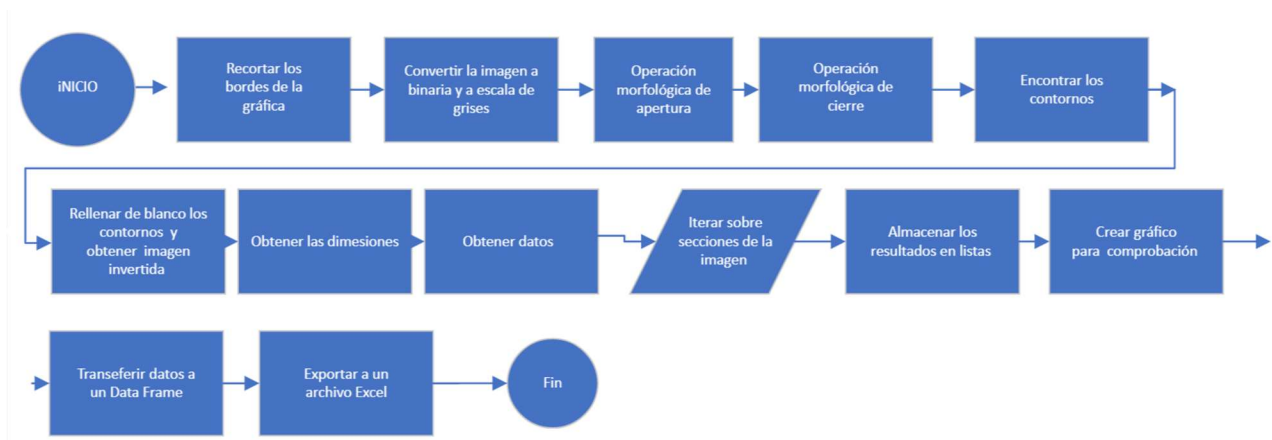


asegura que la imagen se divide en rectángulos de igual tamaño tanto vertical como horizontalmente [20 y 21].

El propósito de este proceso de iteración es dividir la imagen en regiones más pequeñas para obtener sus datos de cada una de ellas.

9. Crear un gráfico con los datos obtenidos, esto para la comprobación de que fuesen correctos.

10. Transferir los datos a un *DataFrame* de *Pandas* y exportar a un archivo Excel.



*Ilustración 5. Procedimiento del primer código de Python*

Posteriormente, se procedió a la identificación de las gráficas faltantes correspondientes a los cuatro sensores, las cuales presentaban similitudes en cuanto al fondo, caracterizado por rectángulos negros punteados y sombras grises en el área de interés, donde se encontraba la línea negra no lineal. El proceso de procesamiento para la eliminación de estas características no deseadas se desarrolló de la siguiente manera [Ilustración 6]:

1. Se realizó la iteración en cada gráfica proporcionada por cada sensor (cuatro por paciente).
2. Se llevó a cabo el recorte de la imagen para eliminar el borde negro de la gráfica, utilizando el tamaño de la imagen y sustrayendo los píxeles a recortar.
3. La imagen fue convertida a escalas de grises para facilitar su procesamiento subsiguiente.
4. Se establecieron umbrales para la detección de sombras grises en la gráfica.
5. Se creó una máscara binaria que identifica los píxeles de la imagen original que están dentro del rango de colores grises definido por los umbrales del paso anterior.

Esta máscara binaria consiste en una matriz de píxeles que tienen solo dos valores posibles, comúnmente 0 y 255, que representan negro y blanco respectivamente. En el contexto de la manipulación de imágenes, una máscara binaria se utiliza para indicar qué píxeles de una imagen original deben ser modificados, resaltados o eliminados.

Se explicará la creación de la máscara binaria en el código con algunos conceptos matemáticos básicos:

Definición de umbrales:

Los umbrales "lower\_gray" y "upper\_gray" especifican el rango de valores de píxeles que se consideran como "grises" en la imagen original. Estos valores están definidos en el espacio de color BGR (Blue-Green-Red).

Operación de umbralización:

La operación cv2.inRange() realiza la umbralización en la imagen original utilizando los umbrales especificados. Esto significa que cada píxel en la imagen original se compara con los valores de umbral. Si el valor del píxel está dentro del rango especificado, se establece en blanco (255); de lo contrario, se establece en negro (0). Matemáticamente, esto se puede representar como:

$$M(x, y) = \begin{cases} 255 & \text{if } L \leq I(x, y) \leq U \\ 0 & \text{otherwise} \end{cases}$$

Donde:

$M(x, y)$  es el valor del píxel en la posición  $(x, y)$  en la máscara binaria.

$I(x, y)$  es el valor del píxel en la posición  $(x, y)$  en la máscara binaria.

$L$  es el valor del umbral inferior.

$U$  es el valor del umbral superior.

Máscara binaria resultante:

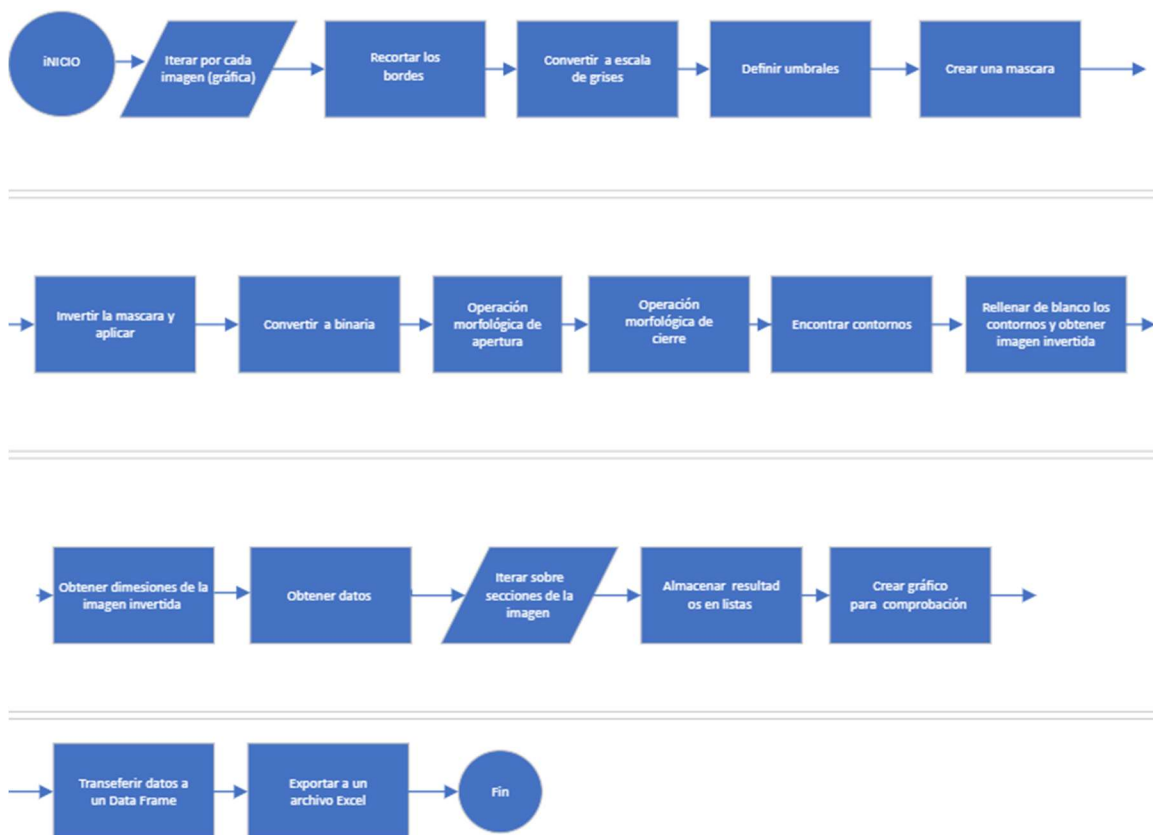
La máscara binaria (gray\_mask) contiene píxeles blancos donde los valores de píxeles en la imagen original están dentro del rango de grises especificado por los umbrales, y píxeles negros en otras ubicaciones.

En resumen, la máscara binaria se crea aplicando un umbral a la imagen original para identificar los píxeles que se consideran "grises", y luego se utiliza esta máscara para aplicar efectos o modificaciones selectivas a la imagen original.

6. Se creó una máscara invertida a partir de la máscara de colores grises. Esto significa que los píxeles que están en blanco en la máscara original se convierten en negros y viceversa. Se utiliza la máscara invertida para aplicar un efecto en la imagen original. En este caso, los píxeles en la imagen original que no son grises (identificados por los píxeles negros en la máscara invertida) se cambian a blanco (255,255,255) en la imagen resultante.
7. La imagen fue nuevamente convertida a escalas de grises y binarizada para su procesamiento adicional.
8. Se aplicó la operación morfológica de apertura para eliminar las líneas punteadas, ajustando el kernel según lo requerido.
9. Se implementó la operación morfológica de cierre en la imagen.
10. Se identificaron los contornos en la imagen cerrada.
11. Los contornos encontrados fueron rellenados de blanco, obteniendo así la imagen invertida.

12. Se extrajeron las dimensiones de la imagen invertida.
13. Se definieron dimensiones del rectángulo.
14. Se llevó a cabo el procesamiento para obtener datos de la imagen invertida.
  - Se iteró sobre secciones de la imagen, procesando cada sección y almacenando los resultados en listas.
15. Se creó un gráfico con los datos obtenidos para verificar su precisión.
16. Los datos se transfirieron a un DataFrame por cada gráfica.
17. Todos los DataFrames fueron concatenados en uno solo.
18. El DataFrame resultante se exportó a un archivo de Excel.

Finalmente, a través de una función, se unificó el código previamente creado para el procesamiento de la gráfica "Índice Heart Logic" con el segundo código referente a las gráficas faltantes, con el objetivo de obtener un único archivo de Excel que contenga los datos de las cinco gráficas.



*Ilustración 6. Diagrama del segundo código de Python.*

### 6.3 Identificación de datos

De manera consecutiva, se detectó que los datos extraídos en Excel carecían de distinción alguna, lo cual generaría un conflicto al visualizarlos y transferirlos a Tableau. Con el fin de abordar esta problemática, se

diseñó un nuevo código que se ajustara a las premisas de identificación de los pacientes, el tipo de gráfica, el tipo de muestra y los límites inferior y superior para cada tipo de identificación de gráfica [Ilustración 7].

Debido a las disparidades encontradas entre los límites presentes en el archivo PDF original y aquellos extraídos en el programa Excel, fue necesario incorporar en el código consultas sobre los límites superiores e inferiores de cada gráfica. Este ajuste se efectuó para asegurar la coherencia entre los parámetros en el archivo resultante y los originales del PDF, facilitando así la generación de alertas al utilizar Tableau.

A continuación, se describen los pasos del código implementado:

1. Importación de librerías.
2. Lectura del documento de Excel generado por el código previo.
3. Designación del nombre de la primera columna como "Serie" y consulta del número de serie, que identifica el archivo PDF y al paciente.
4. Repetición de los datos de la primera columna para el número total de filas en el documento.
5. Ajuste de la longitud de los datos de la primera columna para que coincidan con el número total de filas.
6. Asignación del nombre de la segunda columna como "Tipo".
7. Asignación del nombre de la tercera columna como "Muestra".
8. Asignación del nombre de la cuarta columna como "Límite inferior".
9. Asignación del nombre de la quinta columna como "Límite superior".
10. Creación de listas vacías para almacenar los datos de cada columna.
11. Establecimiento de un bucle para la consulta y recopilación de datos de la segunda columna, en este caso, se repite cinco veces para los cinco tipos de gráficas (1. Índice, 2. S3, 3. S1, 4. FR, 5. ImpToracica), y también se consulta por sus límites en cada tipo, almacenando todo en sus respectivas listas.
12. Creación de otro bucle para que cada tipo, muestra y límites se repitan 305 veces, que representan el número de datos por gráfica (Tipo), guardando cada dato en su respectiva lista.
13. Creación de las nuevas columnas con los datos consultados anteriormente y asignación de las listas a las columnas correspondientes.
14. Almacenamiento del DataFrame modificado en un nuevo archivo de Excel.

Con el archivo Excel final, se ajustaron los límites superiores según la altura del rectángulo acordada en Python. Esto se calculó mediante una fórmula de Excel, teniendo en cuenta que la altura del rectángulo

para cada tipo es diferente. En el ejemplo se muestra el Tipo "1. Índice", que su altura equivale a 156, y este es el 100%. Se aplicó una regla de tres simple para obtener el valor de cada celda A# en todos los datos:

$$=((A2-MIN(\$A\$2:\$A\$281))*100)/156$$

Se restó el mínimo del conjunto de datos por tipo para corregir el desplazamiento de los datos al graficar, asegurando que iniciaran en 0.



Ilustración 7. Diagrama del tercer código.

#### 6.4 Creación de *Dashboard* para la visualización de datos.

Para la implementación del Dashboard, se optó por utilizar Tableau debido a su capacidad para realizar análisis y visualización de datos, así como su habilidad para relacionar diversas bases de datos.

El proceso se inició mediante la importación de las dos bases de datos pertinentes:

- Base de datos obtenida a través de la implementación del primer RPA (Heart Logic).
- Base de datos obtenida mediante el procesamiento de señales en Python (datos\_HL).

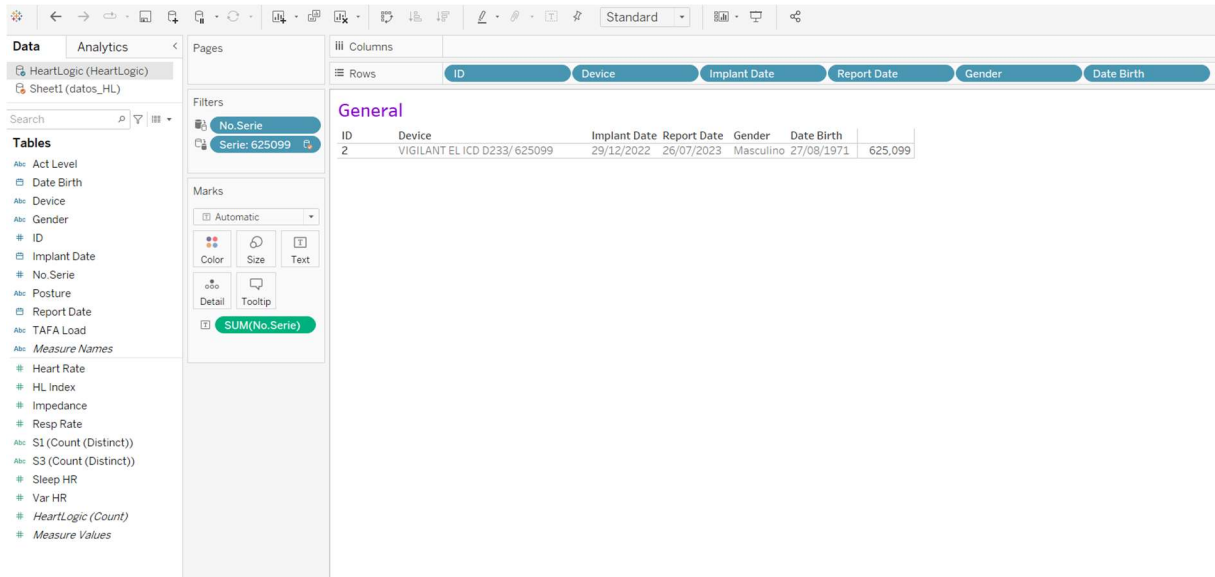
Una vez que ambas bases de datos fueron importadas, se procedió a fusionarlas utilizando el número de serie o el número de paciente como clave de relación, dado que esta vinculación resultaría fundamental para la construcción del panel.

Se determinó que el tablero mostraría visualmente los cinco tipos de gráficas (1. Índice, 2. S3, 3. S1, 4. FR, 5. ImpToracica), junto con la información correspondiente a los dos pacientes y sus respectivos números de serie.

Subsecuentemente, se diseñó la primera tabla que presenta los datos generales del paciente, tales como el dispositivo implantado, la fecha de implantación, la fecha de reporte, el sexo, el año de nacimiento y el

número de serie. Estos datos fueron extraídos mediante el primer RPA y corresponden a la base de datos de *Heart Logic* [Ilustración 8].

Nótese que en la sección de filtros se ha incluido el número de serie, permitiendo así la posibilidad de visualizar la información de manera específica para cada paciente [Ilustración 8].



The screenshot shows a Tableau interface with a table of patient data. The table has the following columns: ID, Device, Implant Date, Report Date, Gender, and Date Birth. A filter for 'No.Serie' is applied, showing a value of 625099. The table contains one row of data for patient ID 2.

| ID | Device                       | Implant Date | Report Date | Gender    | Date Birth |
|----|------------------------------|--------------|-------------|-----------|------------|
| 2  | VIGILANT EL ICD D233/ 625099 | 29/12/2022   | 26/07/2023  | Masculino | 27/08/1971 |

*Ilustración 8. Tabla general con los datos más relevantes del paciente.*

Posteriormente, se llevó a cabo la configuración de la segunda tabla, la cual engloba todos los valores extraídos por sensor mediante el RPA. Estos valores incluyen el Índice Heart Logic, el valor de S3, el valor de S1, la frecuencia cardíaca, la impedancia torácica, la frecuencia respiratoria, la frecuencia cardíaca

nocturna, la variabilidad de la frecuencia cardíaca, la postura al dormir, el nivel de actividad y la carga TA/FA [Ilustración 9]

Nótese que en la sección de filtros se ha vuelto a incluir el número de serie, permitiendo así la posibilidad de visualizar la información de manera específica para cada paciente [Ilustración 9].



*Ilustración 9. Tabla de resultados clínicos por cada sensor del desfibrilador.*

Consecuentemente se edificó la tercera hoja, que alberga las representaciones gráficas por cada tipo, número de serie y mes.

Para determinar el cálculo de los meses incluidos, se consideró un total de 305 datos por tipo de gráfica (cinco tipos en total por cada serie). En el PDF original, estos datos se distribuyen en 6 meses, de modo que una simple división permitió asignar 51 datos a cada mes. Este enfoque facilitó la creación de grupos mensuales en Tableau.

En cada gráfica se incorporó el valor promedio con una línea punteada de color verde [Ilustración 11] y se creó un campo calculado denominado "Lim", donde se aplicó una condición que establecía: "Si la variable [Tipo] es igual a '1. Índice', entonces establecer el límite en 16". Este enunciado significa que el tipo de gráfica "1. Índice" siempre tendría una línea de color rojo en el umbral de 16 [Ilustración 10]. Como se indicó anteriormente, el límite en el Índice Heart Logic para la detección de insuficiencia cardíaca es de 16. Es crucial incorporar este valor como umbral tanto en la gráfica como en la condición, ya que cualquier valor que supere este umbral será de vital importancia para la identificación de la insuficiencia cardíaca. (Ver Ilustración 11)

En la sección de filtros ubicada en el lado izquierdo, se presentan opciones de filtrado por tipo, número de serie y mes [ilustración 11]

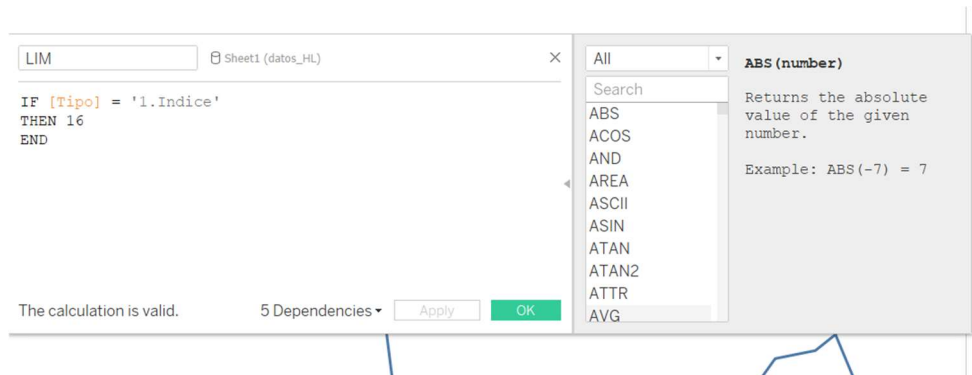


Ilustración 10. Creación del condicional para la visualización del límite en el Índice Heart Logic.

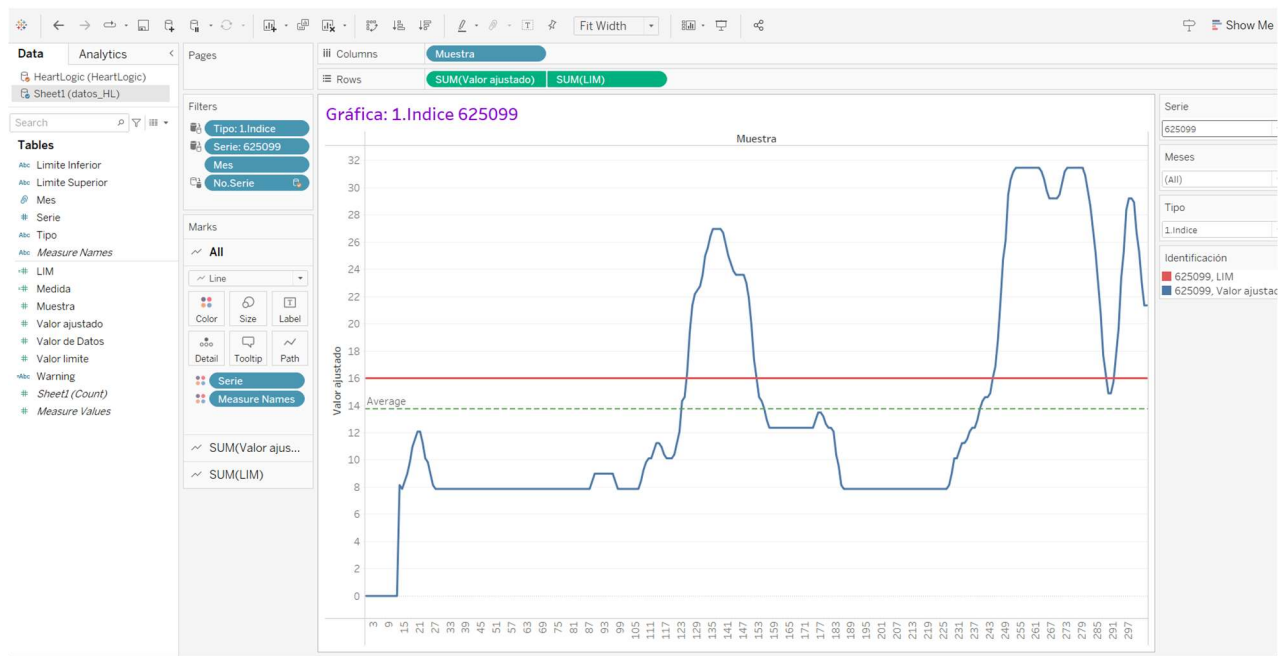


Ilustración 11. Gráfica por el Tipo Índice HL. La línea de color rojo muestra el Límite. La línea de color verde punteada muestra el promedio del conjunto de los datos en un periodo de seis meses.

Finalmente, se implementaron filtros para el tipo, la serie y el mes, y se ajustaron para que se presentaran como listas desplegables, brindando así una interfaz interactiva para el usuario. Además, se incorporaron distintos colores para la serie, el límite del Índice Heart Logic y los valores de los datos "Valor ajustado" [Ilustración 12].



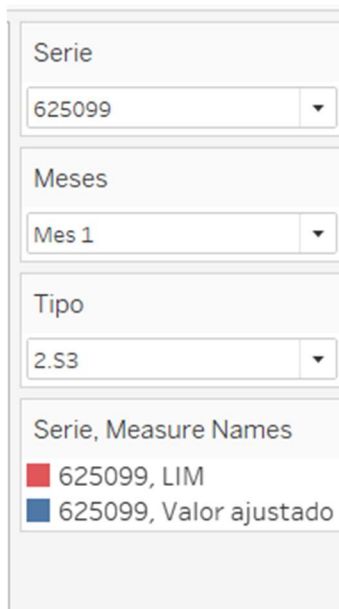


Ilustración 12. Filtros en listas desplegables y colores de referencia para el límite Heart Logic, Serie y los valores de los datos

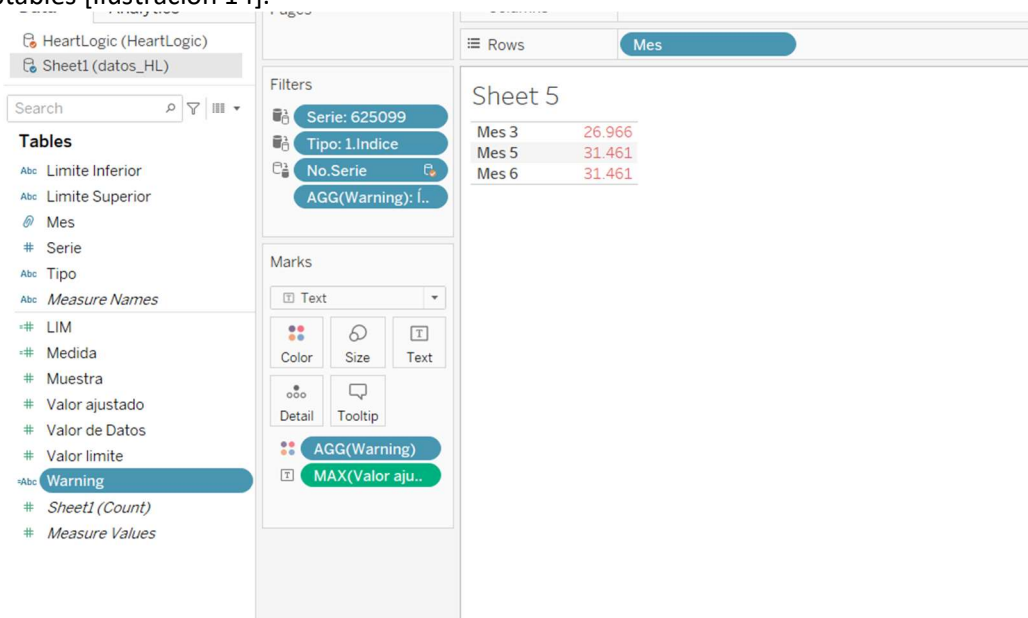
En una hoja adicional de Tableau, se incorporó un campo calculado específicamente para el tipo "1. Índice", con el propósito de establecer una precaución. Esta precaución se manifiesta en la generación de una alerta denominada "Índice HL arriba del límite" en caso de que cualquier valor supere el límite de 16. Además, la alerta indica en qué mes ocurre dicho sobrepaso y proporciona el valor máximo de esos meses que excedieron el límite. Para cualquier otra situación, se presenta la notificación "Índice HL aceptable" [Ilustración 13].



Ilustración 13. Campo calculado con un condicional.

En la última hoja, se diseñó una pequeña tabla que abarca los meses y el valor máximo de los datos de la gráfica del Índice Heart Logic. Se implementó una referencia de colores vinculada al campo calculado

anteriormente, de manera que los valores se presentaran en color rojo si excedían el límite y en verde si eran aceptables [Ilustración 14].



*Ilustración 14. Hoja que involucra una referencia de colores de acuerdo con el campo calculado y una pequeña tabla que contiene los meses con su valor máximo.*

Al concluir, se procedió a la confección del Dashboard unificando todas las hojas previamente elaboradas. Se incorporaron filtros por tipo, serie y mes, los cuales resultan aplicables de manera consistente en todas las hojas. Asimismo, se introdujo un campo calculado que despliega un mensaje acerca del estado del Índice Heart Logic, acompañado de una referencia visual mediante la aplicación de colores. Este enfoque brinda una presentación visual coherente y facilita la interpretación de la información a través de una interfaz interactiva.

## 7 Resultados

### 7.1 Extracción de gráficas y valores por cada sensor

En total, se extrajeron 16 valores de cada paciente a partir de su archivo en formato PDF. Estos valores, obtenidos tanto de los sensores como de datos generales del paciente y del desfibrilador, incluían información como el nombre del dispositivo implantado, la fecha de implantación, el día del reporte, el año de nacimiento y el sexo, este último incorporado manualmente [Ilustración 15].

Con el propósito de salvaguardar la privacidad del paciente, se excluyó su nombre y se le identificó mediante el número de serie del dispositivo. Los 32 valores resultantes fueron organizados en sus respectivas columnas, totalizando 16, y la información de cada archivo se dispuso en filas separadas, sumando 2 filas en total. La tabla resultante fue exportada por el RPA en formato CSV y posteriormente visualizada en Excel [Ilustración 15].

|   | A                              | B            | C           | D        | E    | F    | G         | H         | I        | J           | K         | L        | M          | N      | O          | P         |
|---|--------------------------------|--------------|-------------|----------|------|------|-----------|-----------|----------|-------------|-----------|----------|------------|--------|------------|-----------|
| 1 | Device                         | Implant Date | Report Date | HL Index | S3   | S1   | Impedance | Resp Rate | Sleep HR | Posture     | Act Level | TAFALoad | Heart Rate | Var HR | Date Birth | Gender    |
| 2 | VIGILANT X4 CRT-D G247/ 256410 | 20-oct-22    | 06-jul-23   | 0        | 0,64 | 2,96 | 69        | 18        | 64       | insuficient | 2,0       | 0,0      | 78         | 127    | 26/03/1953 | Femenino  |
| 3 | VIGILANT EL ICD D233/ 6250&9   | 29-dic-22    | 26-jul-23   | 13       | 0,78 | 1,56 | 53        | 16        | 58       | insuficient | 0,5       | 0,0      | 66         | 93     | 27/08/1971 | Masculino |
| 4 |                                |              |             |          |      |      |           |           |          |             |           |          |            |        |            |           |

Ilustración 15. Primera base de datos fabricada por RPA, en formato csv.

La extracción de gráficos se llevó a cabo de manera más selectiva, focalizándose en cinco gráficos de interés: el Índice de insuficiencia cardíaca de Heart Logic, S3, S1, Impedancia torácica y frecuencia respiratoria.

La implementación del RPA fue fundamental para el procedimiento de recorte mencionado anteriormente en la metodología, logrando así la extracción exitosa de las coordenadas de origen y las dimensiones de la imagen ( $x_0, y_0, w, h$ ). Estas dimensiones fueron obtenidas mediante el uso de un software llamado Gimp, que posibilitó al RPA identificar la región de corte adecuada para cada gráfico. Los gráficos resultantes se almacenaron en carpetas específicas correspondientes al nombre del reporte y su tipo.

El código produjo como resultado la creación de 10 carpetas, 5 para cada tipo de gráfica. En la Ilustración 16 y 17, se destaca en verde el nombre del archivo por paciente (HF 06.pdf - Adobe Acrobat Reader (32-bit)), mientras que en amarillo se indica el tipo de gráfica (Índice). Estos elementos, combinados, conforman el nombre de la carpeta.

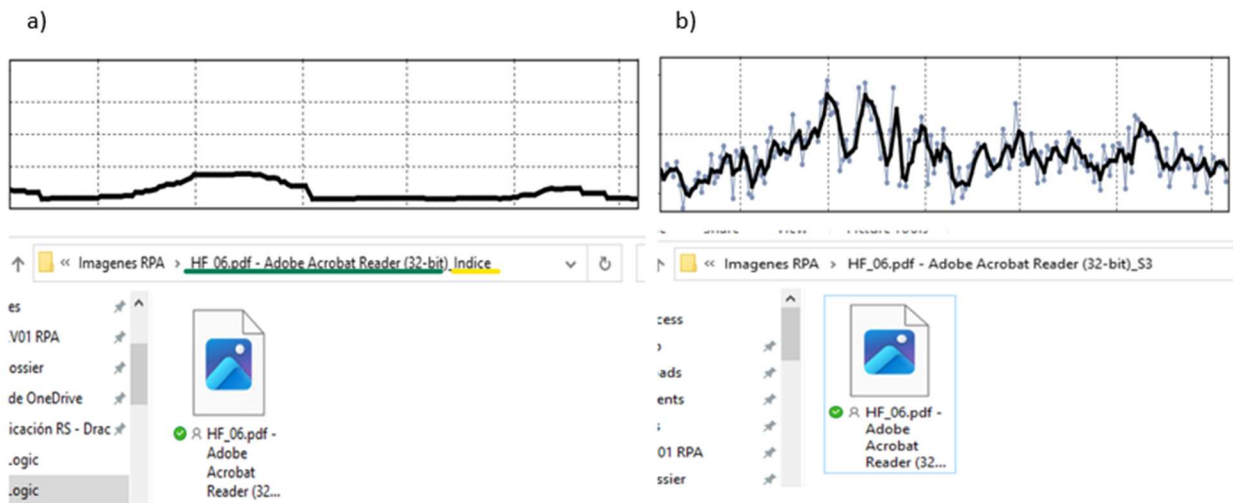


Ilustración 16. a) En la parte superior, el resultado final de la extracción del gráfico índice HL en el primer paciente. En la parte inferior de la imagen se observa el nombre de la carpeta con su gráfico en formato png. b) Formato idéntico, pero sustituyendo el tipo de gráfico por S3.

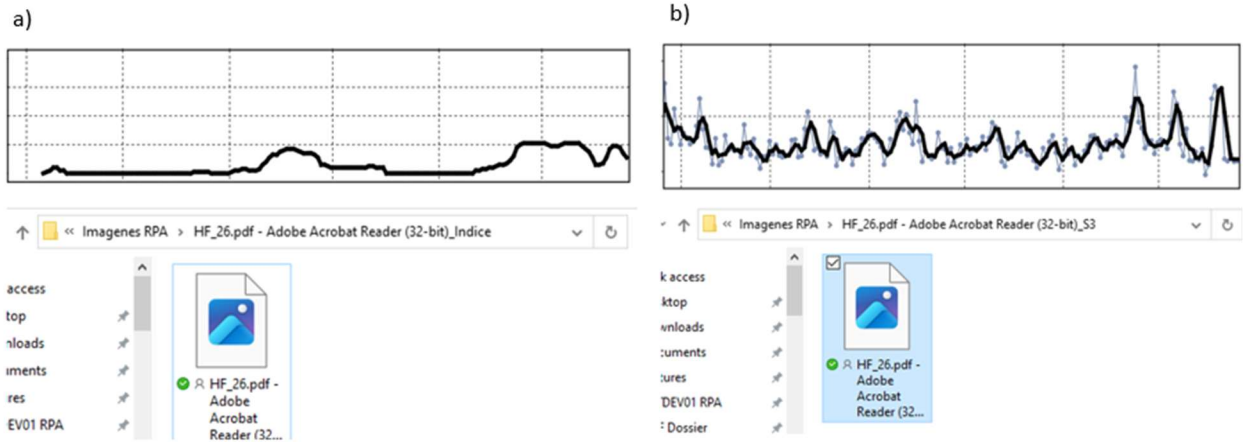


Ilustración 17. Extracción de gráficos a) de tipo índice y b) de tipo S3 para el segundo paciente, junto con sus respectivos archivos en formato PNG.

En la Ilustración 18 el producto resultante de la ejecución del RPA dio lugar a la obtención de diez gráficos, cada uno de ellos almacenado en formato de imagen y asignado a su correspondiente carpeta. En este contexto, se destaca que, con fines de identificación, el nombre de la carpeta para el primer paciente se presenta subrayado en color rojo, mientras que para el segundo paciente se visualiza subrayado en color morado. Esta distinción proporciona una clasificación visual clara y facilita la organización y recuperación de los datos gráficos asociados a cada paciente en particular.

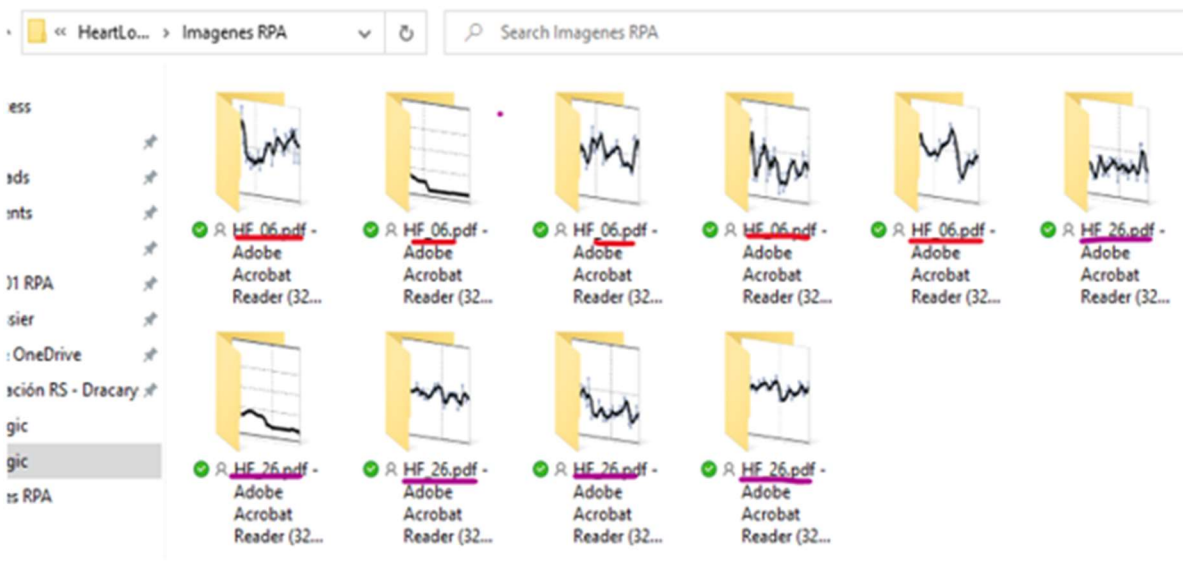


Ilustración 18. Resultado de los diez gráficos extraídos por el RPA.

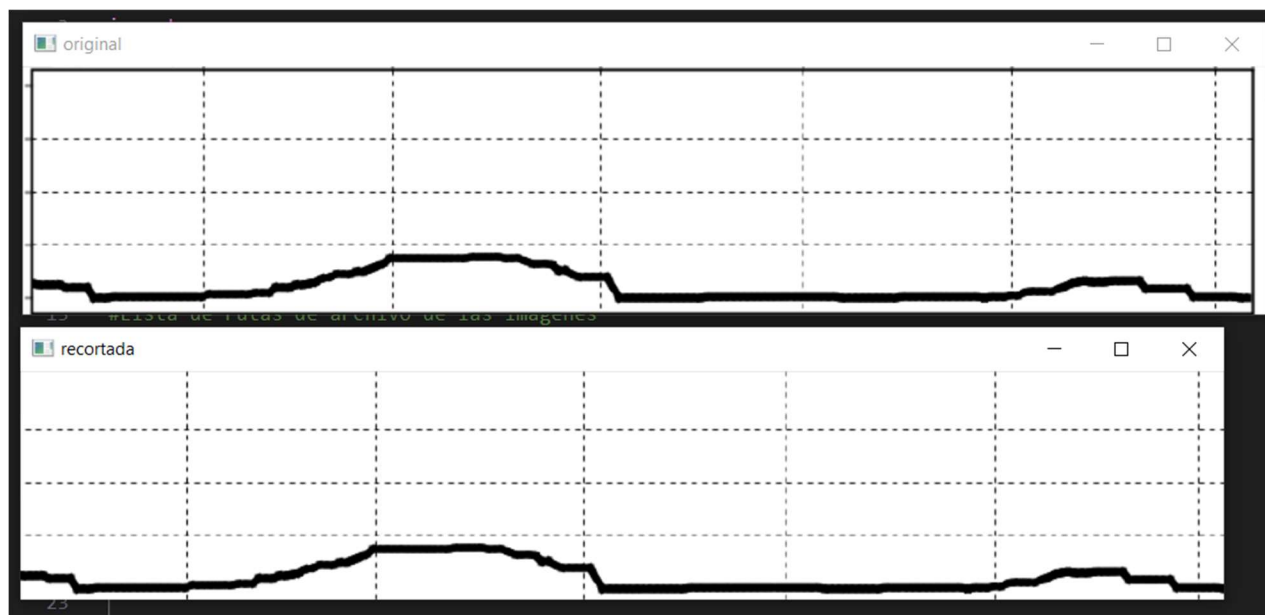
## 7.2 Procesamiento de señales y extracción de datos

De acuerdo con las categorías de los gráficos, se exponen en primer lugar los resultados derivados del procesamiento de la señal y la extracción de datos correspondientes al gráfico del Índice Heart Logic para el primer paciente.

### Primer código

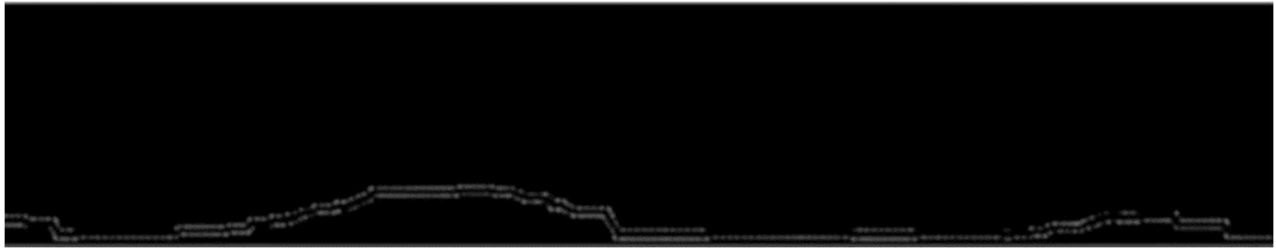
El primer código implementado para este propósito se ejecuta de manera análoga para el segundo paciente, dado que la cantidad de pacientes es de dos. Sin embargo, con el objetivo de facilitar la comprensión, en los resultados se presenta únicamente la información relativa al primer paciente.

El resultado obtenido tras el recorte de la gráfica del Tipo Índice Heart Logic, con el propósito de eliminar el borde negro. Consistió en la resta de los píxeles necesarios al tamaño original de la imagen. Este proceso se llevó a cabo con el objetivo de mejorar la visualización y precisión de los datos extraídos [Ilustración 19].



*Ilustración 19. Resultado del primer paso del código. En la parte superior se observa la gráfica original del índice HL extraída por RPA y en la parte inferior se observa la gráfica recortada sin borde negro.*

Como respuesta a la imagen sin borde negro, se implementó una operación morfológica de apertura. En la ilustración 20, se demuestra cómo esta operación logra eliminar por completo el fondo de líneas negras punteadas, conservando únicamente la línea de interés correspondiente al "Índice Heart Logic".



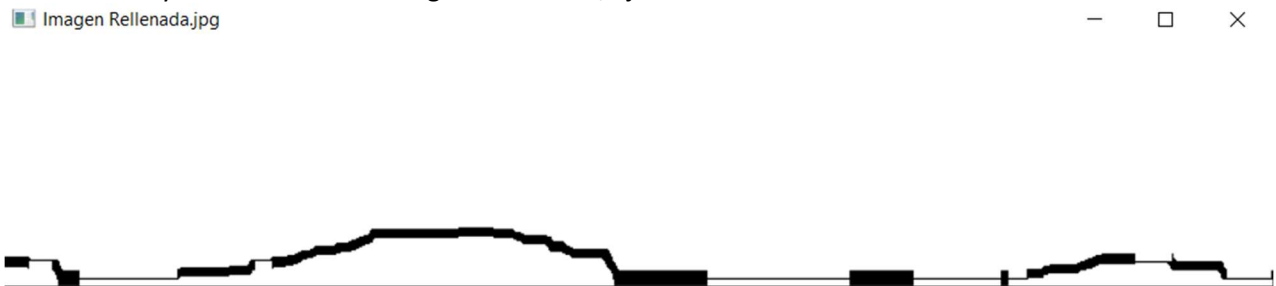
*Ilustración 20. Imagen procesada con la operación morfológica de apertura para eliminar líneas negras punteadas.*

A partir del procesamiento de la imagen con la operación morfológica, se prosiguió a aplicar la operación morfológica de cierre. En la [Ilustración 21] se muestra cómo se logran rellenar los huecos, cabe mencionar que este relleno no es de un 100% debido al ruido que presenta la imagen y al kernel que se usó.



*Ilustración 21. Imagen procesada con la operación morfológica de cierre, para rellenar los huecos generados.*

Después de completar la imagen, se procedió a identificar los contornos. Sin embargo, como se evidencia en las representaciones previas, el fondo se presenta en tono negro, mientras que la línea de interés exhibe una combinación de blanco y gris. En consecuencia, se adoptó la medida de invertir la imagen, transformando el fondo a blanco y la línea de interés a negro, para homologarla con la representación gráfica original [Ilustración 22]. Este proceso de inversión se llevó a cabo con el objetivo de mejorar la visualización y coherencia de la imagen resultante, ajustándola a los estándares deseados.



*Ilustración 22. Imagen Rellenada e invertida después de encontrar los contornos.*

Tras la inversión de la imagen, se procedió a realizar iteraciones a lo largo de la representación gráfica con el fin de recopilar datos específicos de cada sección [Ilustración 23a]. La cantidad total de iteraciones se

estableció en 240 para abarcar de manera exhaustiva la totalidad de la gráfica. Posteriormente, se llevó a cabo la representación gráfica de los datos obtenidos en cada iteración, permitiendo así una visualización más detallada y sistemática de la información.

La validación de los datos recopilados se realizó al comparar la imagen resultante de la representación gráfica con la imagen original. Esta comparación permitió verificar la precisión y consistencia de los datos obtenidos a lo largo de las iteraciones. La concordancia entre la imagen graficada y la original confirmó la fiabilidad del proceso de recopilación de datos, fortaleciendo la confianza en la exactitud de la información obtenida.

En el eje  $x$ , se graficaron las dimensiones de ancho de la imagen y el ancho del rectángulo de interés, mientras que en el eje  $y$  se representaron los datos obtenidos en cada iteración [Ilustración 23b].

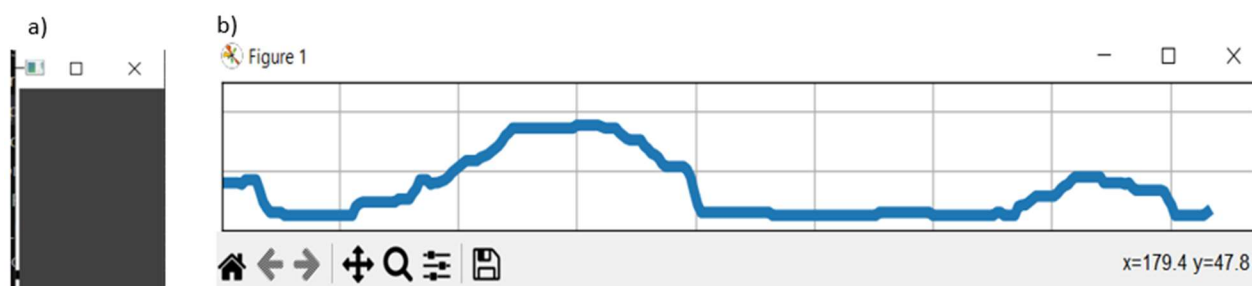
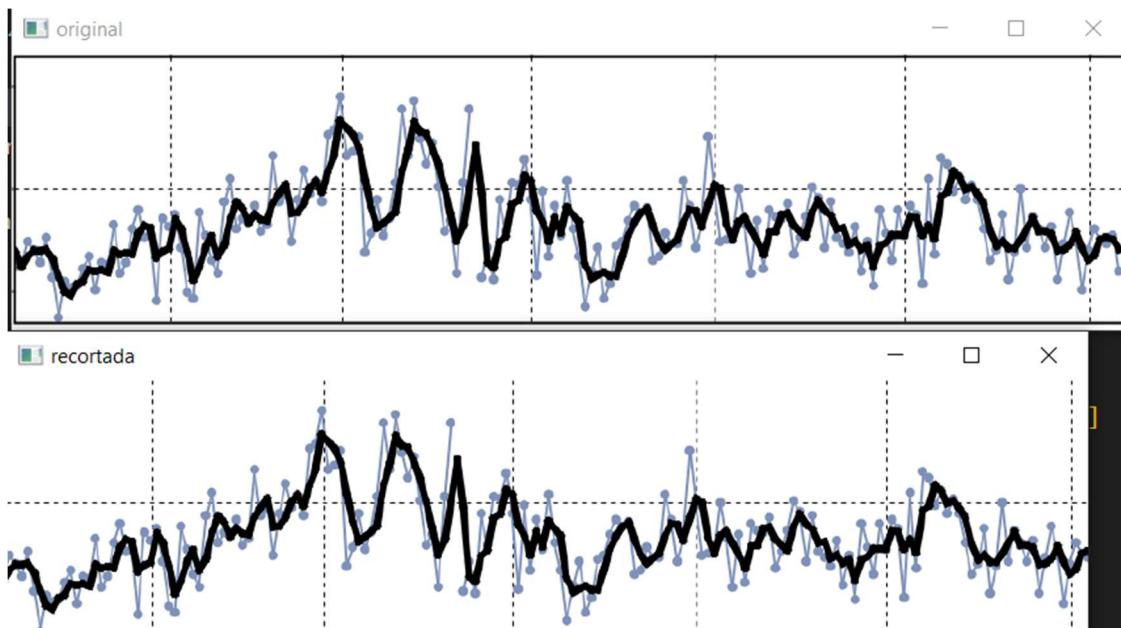


Ilustración 23. a) Corte de un rectángulo para las 240 muestras, estas iteran sobre toda la gráfica para obtener los datos. b) Gráfica de comprobación de los datos extraídos en los pasos anteriores.

## Segundo código

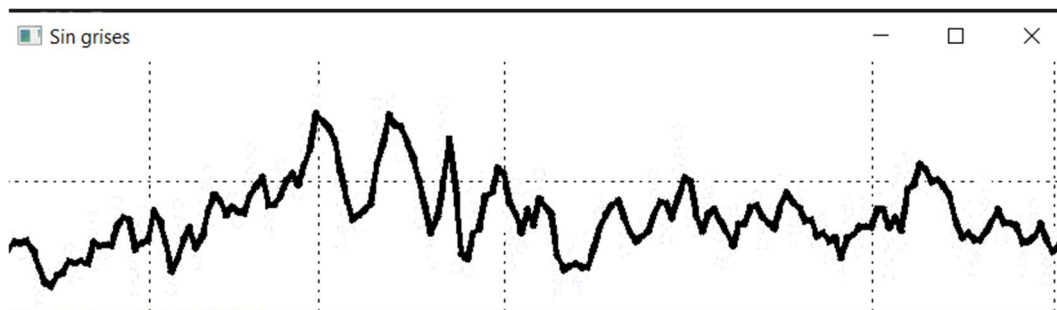
Los resultados del procesamiento de la señal y la extracción de datos se han obtenido para los sensores de Tipo (S3, S1, Impedancia torácica y frecuencia respiratoria). Con el objetivo de facilitar la comprensión, las ilustraciones proporcionadas representan exclusivamente un tipo de gráfica, en este caso, "S3". Es importante señalar que el mismo procedimiento se aplicó de manera iterativa para cada tipo de gráfica y paciente, mediante un bucle que se repite.

El resultado del recorte de la gráfica correspondiente al Tipo S3 tiene como objetivo eliminar el borde negro. Se llevó a cabo una operación de resta de píxeles para ajustar el tamaño recortado al tamaño original de la imagen [Ilustración 24].



*Ilustración 24. Resultado del primer paso del código. En la parte superior se observa la gráfica original del sensor S3 y en la parte inferior, la gráfica recortada sin borde negro.*

Se consiguió suprimir los puntos y tonalidades grises circundantes a la línea negra de interés mediante la implementación de una máscara [Ilustración 25]. Dicha máscara se define como un mapa que especifica qué píxeles deben ser preservados y cuáles deben ser excluidos.



*Ilustración 25. Gráfica procesada sin tonalidades grises, utilizando máscaras.*

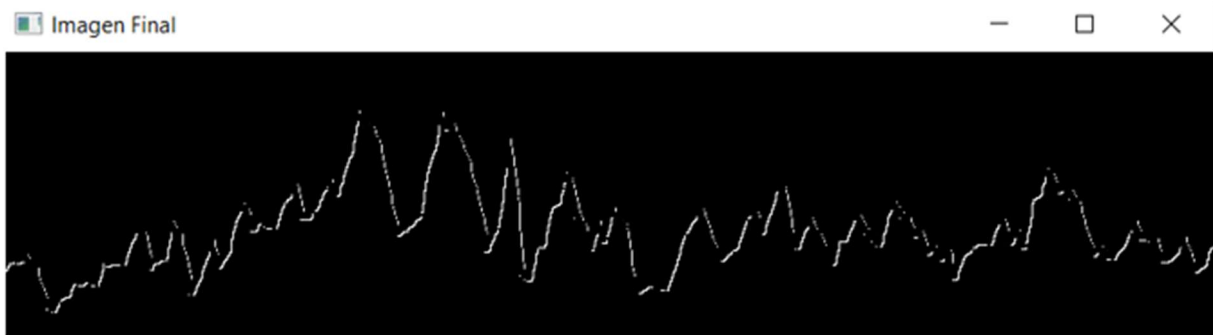
Mediante la implementación de la operación morfológica de apertura, se consigue la eliminación del fondo compuesto por líneas negras punteadas. Posteriormente, se procede a aplicar la operación de cierre con el fin de llevar a cabo el relleno de las áreas resultantes [Ilustración 26].

Las operaciones morfológicas, como la apertura y el cierre, son técnicas utilizadas en el procesamiento de imágenes para modificar la forma y estructura de los objetos en una imagen. La apertura consiste en suavizar contornos, eliminando pequeños detalles y reduciendo el ruido. Por otro lado, el cierre se emplea para unir regiones y completar espacios en blanco, rellenando huecos y proporcionando coherencia estructural. Estas técnicas son fundamentales en la mejora y manipulación de imágenes en diversos

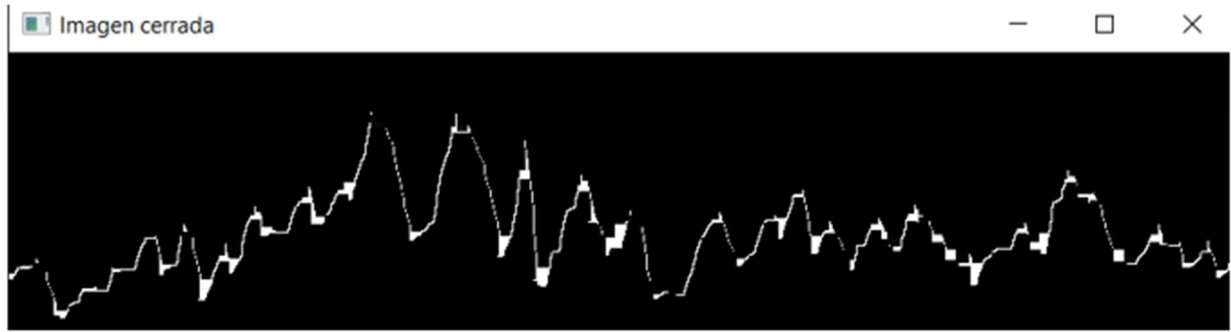


contextos, incluyendo la eliminación de artefactos no deseados y la mejora de la calidad visual.

a)



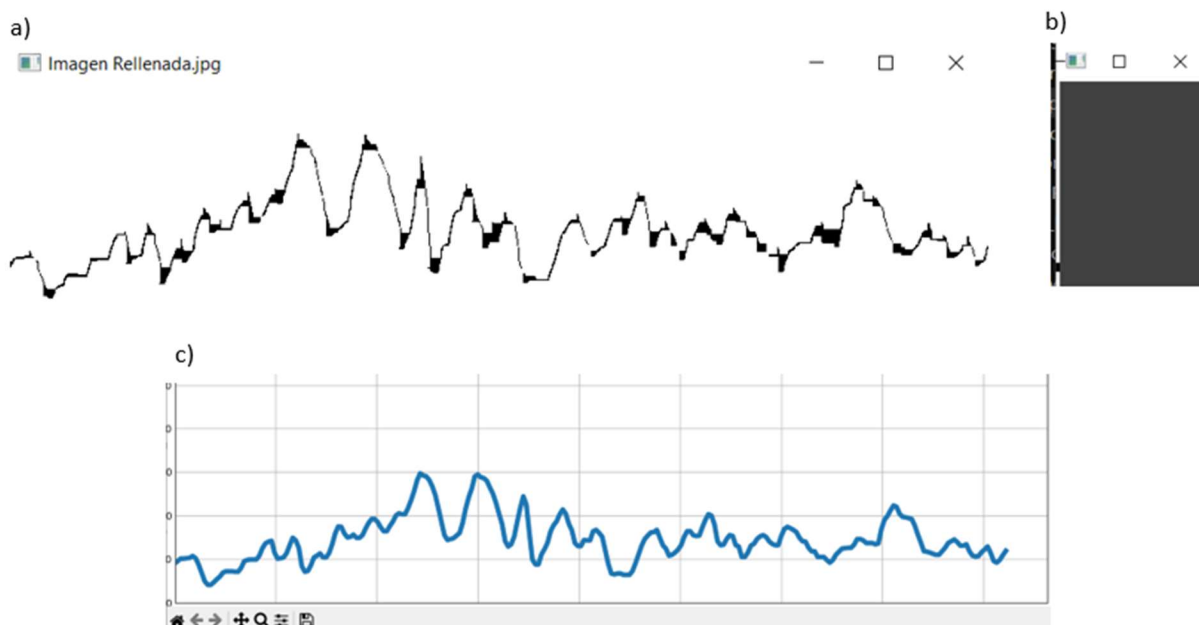
b)



*Ilustración 26.a) Imagen Final procesada con la operación morfológica de open. b) Imagen procesada con la operación morfológica de cierre.*

En una etapa posterior, se identificaron los contornos de la imagen, y se procedió a visualizar la imagen invertida con la reproducción de los colores presentes en la gráfica original [Ilustración 27a]. A continuación, se llevó a cabo la iteración en 240 muestras a lo largo de la imagen con el propósito de recopilar datos,

organizándolos en listas correspondientes [Ilustración 27b]. Por último, se realizó la validación mediante la generación de una gráfica que representara visualmente los datos obtenidos durante el proceso iterativo. En el eje horizontal, se representaron las dimensiones del ancho de la imagen y el ancho del rectángulo, mientras que en el eje vertical se registraron los datos obtenidos en cada iteración [Ilustración 27c].



*Ilustración 27. a) Imagen Rellenada e invertida después de encontrar los contornos. b) En la parte lateral izquierda se aprecia el corte de un rectángulo, este corte itera para las 240 muestras a lo largo de la imagen. c) Gráfica de comprobación obtenida a partir de los datos extraídos.*

Se llevaron a cabo idénticos procedimientos para cada tipo de representación gráfica y para cada paciente. En su totalidad, se procesaron cinco gráficas por paciente, generando un data frame que recopilaba la extracción de datos correspondientes a dichas representaciones gráficas. Cada gráfica resultó en la extracción de 305 valores, los cuales fueron organizados en filas dentro de una hoja de cálculo Excel; consecuentemente, cada archivo individual contenía 1525 filas, equivaliendo a 1525 valores extraídos. Es relevante destacar que para cada número de serie o paciente se generó un archivo de Excel independiente.

### 7.3 Identificación de los datos

#### Tercer código

La identificación de los datos se llevó a cabo siguiendo un protocolo específico que involucra la Serie, el tipo de gráfica, la muestra, así como los límites inferior y superior correspondientes a cada tipo de gráfica. Cada serie, que representa a un paciente, experimentó cinco tipos de extracción de gráficas (1. Índice, 2. S3, 3. S1, 4. FR, 5. ImpT). Para cada tipo de gráfica, se establecieron un límite inferior y un límite superior. Estos elementos, es decir, el tipo de gráfica y los límites, se repitieron consecutivamente para cada conjunto de

305 datos, marcando un cambio en el tipo de gráfica y límites al agotarse dicha cantidad, continuando así hasta abarcar todos los tipos y alcanzar los 1525 datos en total por paciente o serie [Ilustración 28].

Cada número de serie o paciente generó un archivo de Excel único mediante el código correspondiente, resultando en un total de dos archivos. El término "muestra" se refiere a la cuantificación de los datos, que varía de 0 a 305 para cada tipo de gráfica.

Al concluir la identificación de cada número de serie, se consolidaron los dos documentos generados en un solo archivo de Excel, facilitando su manipulación y análisis posterior en plataformas como Tableau.

|    | A              | B           | C         | D       | E               | F               |
|----|----------------|-------------|-----------|---------|-----------------|-----------------|
| 1  | Valor de Datos | Serie       | Tipo      | Muestra | Límite Inferior | Límite Superior |
| 2  |                | 625099      | Índice 0  | 0       |                 | 100             |
| 3  |                | 625099      | Índice 1  | 0       |                 | 100             |
| 4  |                | 625099      | Índice 2  | 0       |                 | 100             |
| 5  |                | 625099      | Índice 3  | 0       |                 | 100             |
| 6  |                | 625099      | Índice 4  | 0       |                 | 100             |
| 7  |                | 625099      | Índice 5  | 0       |                 | 100             |
| 8  |                | 625099      | Índice 6  | 0       |                 | 100             |
| 9  |                | 625099      | Índice 7  | 0       |                 | 100             |
| 10 |                | 625099      | Índice 8  | 0       |                 | 100             |
| 11 |                | 625099      | Índice 9  | 0       |                 | 100             |
| 12 |                | 625099      | Índice 10 | 0       |                 | 100             |
| 13 |                | 625099      | Índice 11 | 0       |                 | 100             |
| 14 |                | 625099      | Índice 12 | 0       |                 | 100             |
| 15 |                | 14.5 625099 | Índice 13 | 0       |                 | 100             |
| 16 |                | 14 625099   | Índice 14 | 0       |                 | 100             |
| 17 |                | 15 625099   | Índice 15 | 0       |                 | 100             |
| 18 |                | 16 625099   | Índice 16 | 0       |                 | 100             |
| 19 |                | 17.5 625099 | Índice 17 | 0       |                 | 100             |
| 20 |                | 18.5 625099 | Índice 18 | 0       |                 | 100             |
| 21 |                | 20.5 625099 | Índice 19 | 0       |                 | 100             |
| 22 |                | 21.5 625099 | Índice 20 | 0       |                 | 100             |
| 23 |                | 21.5 625099 | Índice 21 | 0       |                 | 100             |
| 24 |                | 20 625099   | Índice 22 | 0       |                 | 100             |
| 25 |                | 18 625099   | Índice 23 | 0       |                 | 100             |
| 26 |                | 17.5 625099 | Índice 24 | 0       |                 | 100             |
| 27 |                | 15 625099   | Índice 25 | 0       |                 | 100             |
| 28 |                | 14.5 625099 | Índice 26 | 0       |                 | 100             |
| 29 |                | 14 625099   | Índice 27 | 0       |                 | 100             |
| 30 |                | 14 625099   | Índice 28 | 0       |                 | 100             |
| 31 |                | 14 625099   | Índice 29 | 0       |                 | 100             |
| 32 |                | 14 625099   | Índice 30 | 0       |                 | 100             |
| 33 |                | 14 625099   | Índice 31 | 0       |                 | 100             |
| 34 |                | 14 625099   | Índice 32 | 0       |                 | 100             |
| 35 |                | 14 625099   | Índice 33 | 0       |                 | 100             |
| 36 |                | 14 625099   | Índice 34 | 0       |                 | 100             |
| 37 |                | 14 625099   | Índice 35 | 0       |                 | 100             |
| 38 |                | 14 625099   | Índice 36 | 0       |                 | 100             |
| 39 |                | 14 625099   | Índice 37 | 0       |                 | 100             |
| 40 |                | 14 625099   | Índice 38 | 0       |                 | 100             |
| 41 |                | 14 625099   | Índice 39 | 0       |                 | 100             |
| 42 |                | 14 625099   | Índice 40 | 0       |                 | 100             |
| 43 |                | 14 625099   | Índice 41 | 0       |                 | 100             |
| 44 |                | 14 625099   | Índice 42 | 0       |                 | 100             |
| 45 |                | 14 625099   | Índice 43 | 0       |                 | 100             |
| 46 |                | 14 625099   | Índice 44 | 0       |                 | 100             |
| 47 |                | 14 625099   | Índice 45 | 0       |                 | 100             |
| 48 |                | 14 625099   | Índice 46 | 0       |                 | 100             |
| 49 |                | 14 625099   | Índice 47 | 0       |                 | 100             |
| 50 |                | 14 625099   | Índice 48 | 0       |                 | 100             |
| 51 |                | 14 625099   | Índice 49 | 0       |                 | 100             |
| 52 |                | 14 625099   | Índice 50 | 0       |                 | 100             |
| 53 |                | 14 625099   | Índice 51 | 0       |                 | 100             |
| 54 |                | 14 625099   | Índice 52 | 0       |                 | 100             |
| 55 |                | 14 625099   | Índice 53 | 0       |                 | 100             |
| 56 |                | 14 625099   | Índice 54 | 0       |                 | 100             |
| 57 |                | 14 625099   | Índice 55 | 0       |                 | 100             |
| 58 |                | 14 625099   | Índice 56 | 0       |                 | 100             |
| 59 |                | 14 625099   | Índice 57 | 0       |                 | 100             |
| 60 |                | 14 625099   | Índice 58 | 0       |                 | 100             |
| 61 |                | 14 625099   | Índice 59 | 0       |                 | 100             |
| 62 |                | 14 625099   | Índice 60 | 0       |                 | 100             |

Ilustración 28. Identificación por Serie, Tipo, Muestra y Límites del segundo paciente con número de serie (625099).

En la etapa final, el archivo Excel consolidado que albergaba los datos de ambos pacientes experimentó un ajuste en los límites superiores. Este ajuste se realizó de conformidad con la altura previamente establecida para el rectángulo, la cual fue determinada en el entorno de programación Python.

El cálculo fue mediante la aplicación de la fórmula detallada en la metodología mencionada. En la parte superior de la Ilustración 29 se muestra la fórmula para calcular el valor y en la tabla aparece la columna agregada con el nombre “Valor ajustado”.

B2     $= (A2 - \text{MIN}(SAS2:SAS281)) * 100 / 156$

| Valor de Datos | Valor ajustado | Serie   | Tipo     | Muestra | Límite Inferior | Límite Superior |
|----------------|----------------|---------|----------|---------|-----------------|-----------------|
| 14             | 9.615184615    | 2886410 | 1.Índice | 0       | 0               | 100             |
| 15             | 9.615184615    | 2886410 | 1.Índice | 1       | 0               | 100             |
| 15             | 9.615184615    | 2886410 | 1.Índice | 2       | 0               | 100             |
| 15             | 9.615184615    | 2886410 | 1.Índice | 3       | 0               | 100             |
| 15             | 9.615184615    | 2886410 | 1.Índice | 4       | 0               | 100             |
| 14.5           | 9.29471795     | 2886410 | 1.Índice | 5       | 0               | 100             |
| 16.5           | 10.57692308    | 2886410 | 1.Índice | 6       | 0               | 100             |
| 16             | 10.25641026    | 2886410 | 1.Índice | 7       | 0               | 100             |
| 16             | 10.25641026    | 2886410 | 1.Índice | 8       | 0               | 100             |
| 16             | 10.25641026    | 2886410 | 1.Índice | 9       | 0               | 100             |
| 16             | 10.25641026    | 2886410 | 1.Índice | 10      | 0               | 100             |
| 11.5           | 7.371794872    | 2886410 | 1.Índice | 11      | 0               | 100             |
| 9.5            | 6.08974359     | 2886410 | 1.Índice | 12      | 0               | 100             |
| 6.5            | 4.166666667    | 2886410 | 1.Índice | 13      | 0               | 100             |
| 6              | 3.846153846    | 2886410 | 1.Índice | 14      | 0               | 100             |
| 6              | 3.846153846    | 2886410 | 1.Índice | 15      | 0               | 100             |
| 6              | 3.846153846    | 2886410 | 1.Índice | 16      | 0               | 100             |
| 3.5            | 2.243589744    | 2886410 | 1.Índice | 17      | 0               | 100             |
| 3.5            | 2.243589744    | 2886410 | 1.Índice | 18      | 0               | 100             |
| 3.5            | 2.243589744    | 2886410 | 1.Índice | 19      | 0               | 100             |
| 3.5            | 2.243589744    | 2886410 | 1.Índice | 20      | 0               | 100             |
| 3.5            | 2.243589744    | 2886410 | 1.Índice | 21      | 0               | 100             |
| 3.5            | 2.243589744    | 2886410 | 1.Índice | 22      | 0               | 100             |
| 3.5            | 2.243589744    | 2886410 | 1.Índice | 23      | 0               | 100             |
| 3.5            | 2.243589744    | 2886410 | 1.Índice | 24      | 0               | 100             |
| 3.5            | 2.243589744    | 2886410 | 1.Índice | 25      | 0               | 100             |
| 3.5            | 2.243589744    | 2886410 | 1.Índice | 26      | 0               | 100             |
| 3.5            | 2.243589744    | 2886410 | 1.Índice | 27      | 0               | 100             |
| 3.5            | 2.243589744    | 2886410 | 1.Índice | 28      | 0               | 100             |
| 3.5            | 2.243589744    | 2886410 | 1.Índice | 29      | 0               | 100             |
| 3.5            | 2.243589744    | 2886410 | 1.Índice | 30      | 0               | 100             |
| 3.5            | 2.243589744    | 2886410 | 1.Índice | 31      | 0               | 100             |
| 3.5            | 2.243589744    | 2886410 | 1.Índice | 32      | 0               | 100             |
| 3.5            | 2.243589744    | 2886410 | 1.Índice | 33      | 0               | 100             |
| 3.5            | 2.243589744    | 2886410 | 1.Índice | 34      | 0               | 100             |
| 3.5            | 2.243589744    | 2886410 | 1.Índice | 35      | 0               | 100             |
| 3.5            | 2.243589744    | 2886410 | 1.Índice | 36      | 0               | 100             |
| 3.5            | 2.243589744    | 2886410 | 1.Índice | 37      | 0               | 100             |
| 4.5            | 4.166666667    | 2886410 | 1.Índice | 38      | 0               | 100             |
| 8.5            | 5.448717949    | 2886410 | 1.Índice | 39      | 0               | 100             |
| 8.5            | 5.448717949    | 2886410 | 1.Índice | 40      | 0               | 100             |
| 8.5            | 5.448717949    | 2886410 | 1.Índice | 41      | 0               | 100             |
| 8.5            | 5.448717949    | 2886410 | 1.Índice | 42      | 0               | 100             |
| 8.5            | 5.448717949    | 2886410 | 1.Índice | 43      | 0               | 100             |
| 8.5            | 5.448717949    | 2886410 | 1.Índice | 44      | 0               | 100             |
| 8.5            | 5.448717949    | 2886410 | 1.Índice | 45      | 0               | 100             |
| 8.5            | 5.448717949    | 2886410 | 1.Índice | 46      | 0               | 100             |
| 8.5            | 5.448717949    | 2886410 | 1.Índice | 47      | 0               | 100             |
| 8.5            | 5.448717949    | 2886410 | 1.Índice | 48      | 0               | 100             |
| 9              | 5.769230769    | 2886410 | 1.Índice | 49      | 0               | 100             |
| 9.5            | 6.08974359     | 2886410 | 1.Índice | 50      | 0               | 100             |
| 9.5            | 6.08974359     | 2886410 | 1.Índice | 51      | 0               | 100             |
| 9.5            | 6.08974359     | 2886410 | 1.Índice | 52      | 0               | 100             |
| 10             | 6.41025641     | 2886410 | 1.Índice | 53      | 0               | 100             |
| 11.5           | 7.371794872    | 2886410 | 1.Índice | 54      | 0               | 100             |
| 13             | 8.333333333    | 2886410 | 1.Índice | 55      | 0               | 100             |
| 16             | 10.25641026    | 2886410 | 1.Índice | 56      | 0               | 100             |

Ilustración 29. Columna agregada con el nombre “Valor ajustado”.

## 7.4 Dashboard para la visualización de datos

### 7.4.1 Primera parte del Dashboard Heart Logic

En la primera parte del Dashboard se muestran los siguientes apartados.

- Gráfica que el usuario elija de acuerdo con su Tipo y Serie.
- Promedio de la gráfica de acuerdo con su Tipo, Serie y Mes.
- En la gráfica correspondiente al Tipo ÍNDICE HL, se ha establecido un límite de 16, mostrado por una línea roja. Por consiguiente, cualquier valor que este por arriba de este límite presentara un anuncio en la interfaz indicando “Índice HL arriba del límite”. Este límite se visualiza por Serie y Mes.
- Meses que sobrepasaron el límite HL.

#### Filtros por:

1. Tipo (1. Índice ,2. S3, 3. S1, 4.FR, 5. ImpT)

2. Paciente No. Serie (286410 y 625099)
3. Meses (Mes 1, Mes 2, Mes 3, Mes 4, Mes 5 y Mes 6)
4. Cabe mencionar que los filtros se aplican a todas las partes del tablero.

En la Ilustración 30, se presenta una representación gráfica de Tipo "1. Índice" correspondiente a la Serie 286410. Enmarcada por un rectángulo azul en la parte superior, la gráfica exhibe el promedio mediante una línea verde punteada y un límite señalado por una línea roja con un valor de 16. El eje x del gráfico representa el rango de datos por Tipo, mientras que el eje y, indica el valor asociado a cada dato.

En el centro de la imagen, en el lado derecho, se aprecian filtros disponibles para el usuario. Por último, en la parte inferior del lado derecho, dentro de un rectángulo lila, se presenta el aviso "Índice HL está por encima del límite". Adyacente a este anuncio se detallan los meses en los cuales dicho índice superó el límite, así como el valor máximo alcanzado durante esos meses.

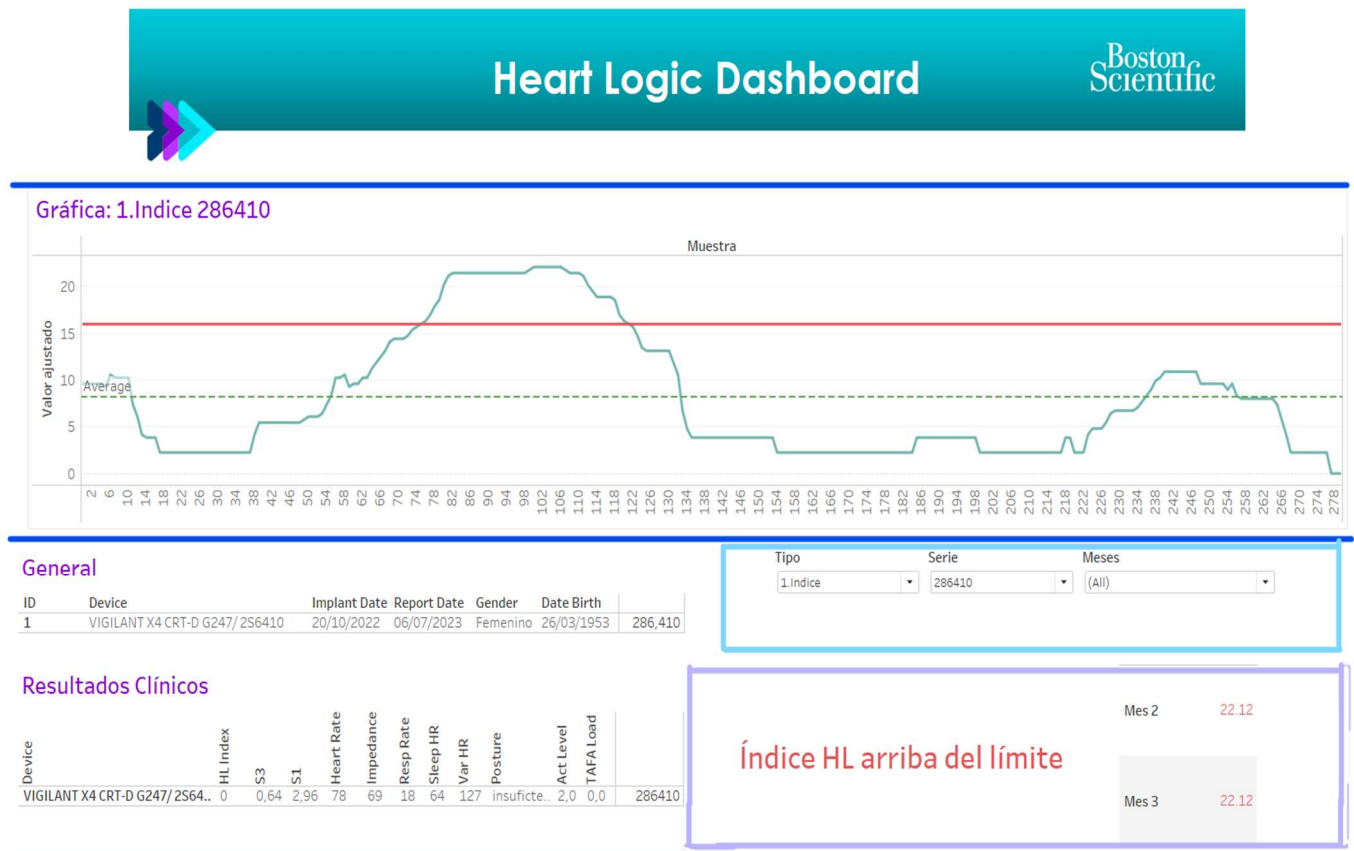


Ilustración 30. Heart Logic Dashboard. Cuadro azul marino, la gráfica de acuerdo con su tipo. Cuadro azul cielo, los filtros que se pueden aplicar. Cuadro lila, la alerta con sus respectivos datos.

### 7.4.2 Segunda parte del *Dashboard Heart Logic*

Este comprende la primera base de datos extraída por el Proceso de Automatización Robótica (RPA), la cual engloba información integral del paciente, incluyendo el identificador (ID), el dispositivo empleado, la fecha de implante, la fecha del informe, el género y el año de nacimiento. Con el propósito de salvaguardar la privacidad del paciente, se empleó el número de serie en lugar de su nombre [Ilustración 31].

Asimismo, abarca los resultados clínicos correspondientes al paciente, tales como el índice HL, el valor de los sensores S3, S1, la frecuencia cardíaca, así como su cálculo nocturno, la impedancia torácica, la frecuencia respiratoria, la postura durante el sueño, el nivel de actividad, la carga de TA/FA y la variabilidad de la frecuencia cardíaca [Ilustración 31]. Cabe destacar que todos estos parámetros se refieren exclusivamente al día de descarga del informe.

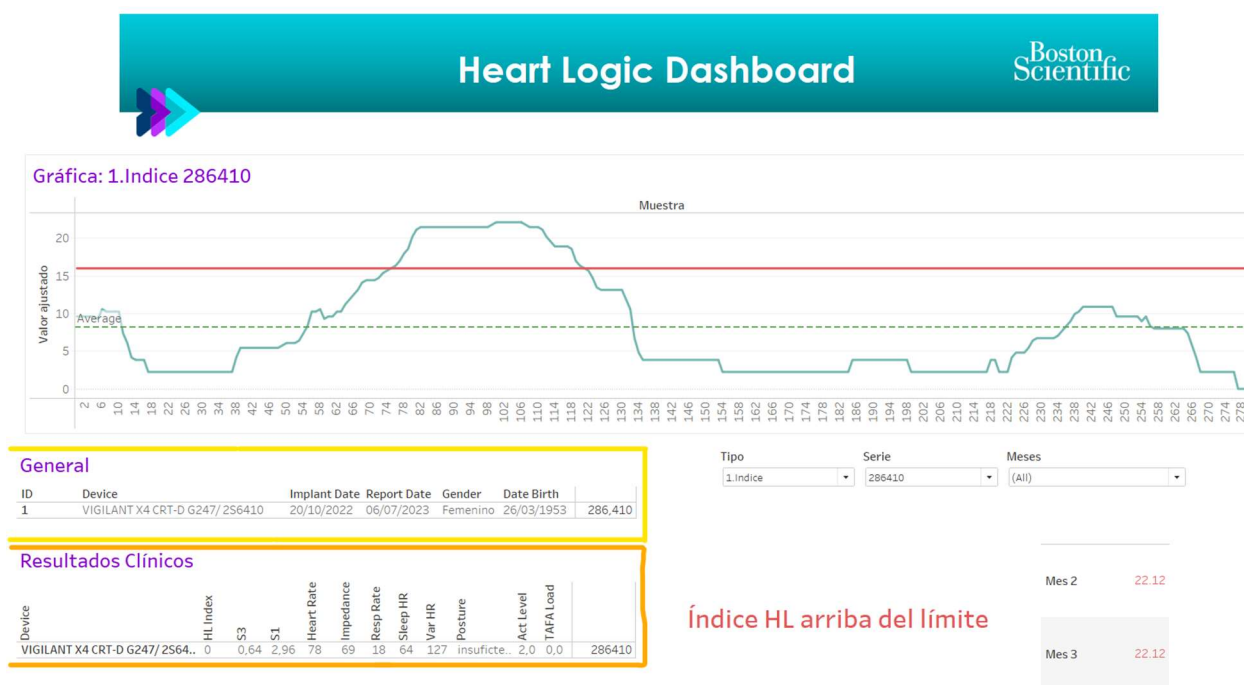
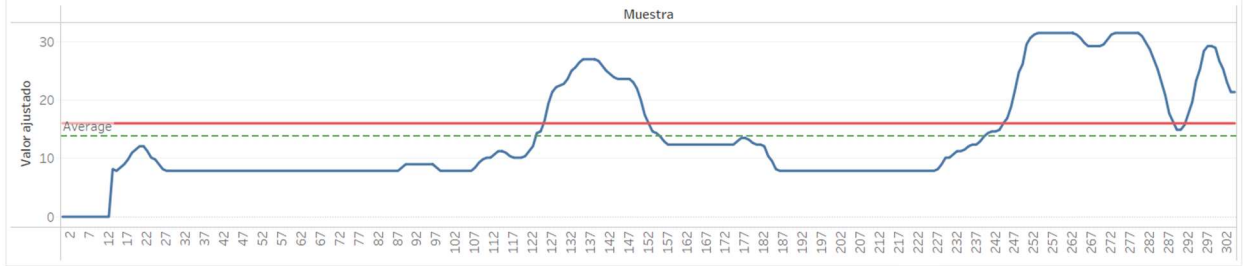


Ilustración 31. *Heart Logic Dashboard*. En la parte central de lado izquierdo con un rectángulo amarillo se muestran los datos generales del paciente con número de serie 286410. En la parte inferior del lado izquierdo con un rectángulo naranja se observan los resultados clínicos de ese mismo paciente.

El Heart Logic Dashboard para el paciente con el número de serie 625099 se presenta de manera similar [Ilustración 32], destacando que se ha aplicado un filtro para exhibir la información del segundo paciente. Cada componente del tablero guarda similitud estructural, variando únicamente los valores específicos. Es importante señalar que, al igual que el paciente anterior, este también ha superado el umbral del límite, por lo que persiste la presencia de un anuncio relacionado con el Índice HL.

Gráfica: 1.Indice 625099



**General**

| ID | Device                       | Implant Date | Report Date | Gender    | Date Birth |         |
|----|------------------------------|--------------|-------------|-----------|------------|---------|
| 2  | VIGILANT EL ICD D233/ 625099 | 29/12/2022   | 26/07/2023  | Masculino | 27/08/1971 | 625,099 |

Tipo:  Serie:  Meses:

**Identificación**  
■ 625099, LIM  
■ 625099, Valor ajustado

**Resultados Clínicos**

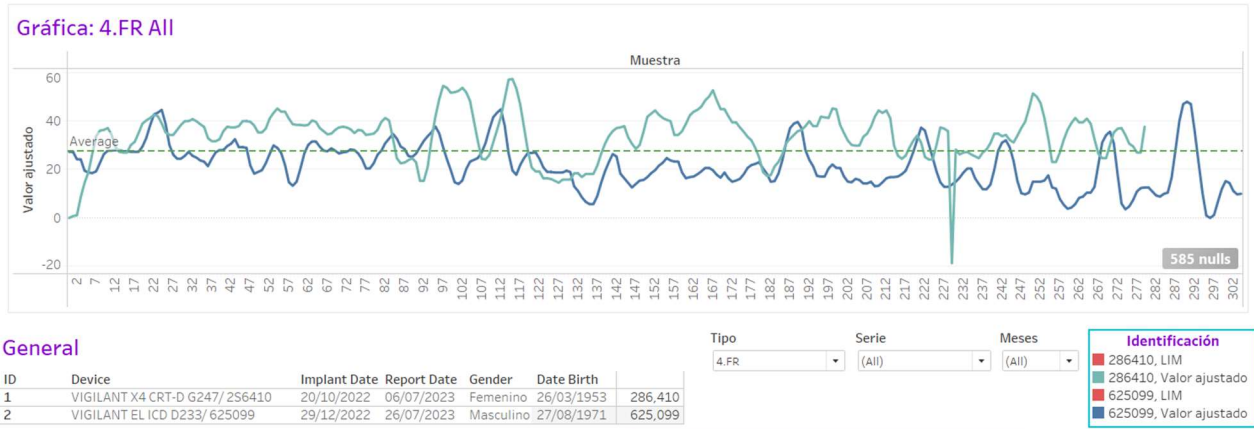
| Device                       | HL Index | S3   | S1   | Heart Rate | Impedance | Resp Rate | Sleep HR | Var HR | Posture     | Act Level | TAFALoad |
|------------------------------|----------|------|------|------------|-----------|-----------|----------|--------|-------------|-----------|----------|
| VIGILANT EL ICD D233/ 625099 | 13       | 0.78 | 1.56 | 66         | 53        | 16        | 58       | 93     | insuficte.. | 0.5       | 0,0      |

Índice HL arriba del límite

|       |        |
|-------|--------|
| Mes 3 | 26.966 |
| Mes 5 | 31.461 |
| Mes 6 | 31.461 |

*Ilustración 32. Heart Logic Dashboard para diferente paciente con número de serie 625099.*

El Heart Logic Dashboard[Ilustración 33], al aplicar los filtros correspondientes al Tipo "4.FR" y considerando los dos pacientes identificados por sus respectivos números de serie, presenta una gráfica que incorpora ambas series. En la sección central del lado derecho, resaltada por un rectángulo azul, se observa una codificación de colores para identificar cada paciente. Tanto los datos generales como los resultados clínicos difieren entre ambos pacientes. En este escenario, al no haber filtrado por el Tipo "Índice", el anuncio se muestra como "Índice HL aceptable" dado que se refiere a otro Tipo y no se encuentra dentro de los parámetros filtrados.



**Resultados Clínicos**

| Device                         | HL Index | S3   | S1   | Heart Rate | Impedance | Resp Rate | Sleep HR | Var HR | Posture     | Act Level | TAFALoad |
|--------------------------------|----------|------|------|------------|-----------|-----------|----------|--------|-------------|-----------|----------|
| VIGILANT EL ICD D233/ 625099   | 13       | 0,78 | 1,56 | 66         | 53        | 16        | 58       | 93     | insuficte.. | 0,5       | 0,0      |
| VIGILANT X4 CRT-D G247/ 2S64.. | 0        | 0,64 | 2,96 | 78         | 69        | 18        | 64       | 127    | insuficte.. | 2,0       | 0,0      |

**Índice HL aceptable**

*Ilustración 33. Heart Logic Dashboard aplicando los filtros de Tipo "4.FR" y los dos pacientes con su número de Serie. El cuadro de identificación es por colores e indica el número de paciente.*

**8 Conclusiones**

A través de la extracción, análisis y procesamiento de señales mediante el uso de Python y la automatización proporcionada por "Automation Anywhere", se ha creado y consolidado una base de datos destinada al almacenamiento y visualización de información proveniente de sensores y del índice Heart Logic.

La formación de esta base de datos facilita la preservación a largo plazo de información representativa de cada paciente, permitiendo un acceso sencillo y eficiente a los datos. Con la base de datos procesada, se presenta una herramienta de visualización que posibilita a los médicos especialistas examinar la demografía del paciente, visualizar gráficas mensuales de los cinco sensores, revisar valores de otros sensores, identificar alertas de irregularidades y anticipar eventos clínicos mayores.

El Dashboard Heart Logic constituye un avance significativo para la detección de irregularidades a lo largo del tiempo de uso del desfibrilador. Esto capacita al médico para realizar análisis predictivos frente a sospechas de complicaciones futuras que requieran atención especializada en enfermedades cardiacas. Asimismo, permite la comparación entre dos o más pacientes para realizar un análisis más preciso. Además de visualizar datos específicos y posibles alertas en intervalos más cortos, como una semana específica dentro de los seis meses analizados en la muestra semestral.



Con la incorporación de más pacientes y la acumulación de datos, se plantea la posibilidad de realizar análisis descriptivos-predictivos en el futuro. El uso de datos históricos y actuales podría emplearse para prever enfermedades cardíacas mediante la aplicación de técnicas de aprendizaje automático. Este enfoque implicaría entrenar algoritmos para interpretar conjuntos de datos históricos, identificar tendencias, realizar predicciones y simular resultados.

Este proyecto aportó un gran valor a la empresa, ya que se podía replicar en diferentes áreas con productos variados, no solo en el ámbito cardiovascular. Desde mi perspectiva, fue un proyecto en el que experimenté un crecimiento profesional significativo. Aprendí a definir prioridades, a comunicarme con personas relevantes, a mapear los pasos necesarios para automatizar, definir procesos y optimizarlos y, finalmente, a desarrollarlo. En este proceso mejoré mis habilidades técnicas de programación y de análisis de datos.

## 9 Bibliografía

1. ¿Qué es la insuficiencia cardíaca? (s. f.). National Institute on Aging. <https://www.nia.nih.gov/espanol/corazon/insuficiencia-cardiaca>
2. Mozaffarian D, Benjamin EJ, Go AS, et al. Heart disease and stroke statistics-2016 update: a informe de la Asociación Americana del Corazón. *Circulation* 2016;133:e38-360.
3. Yancy CW, Jessup M, Bozkurt B, et al. 2013 ACCF/AHA guideline for the management of heart failure: a report of the American College of Cardiology Foundation/American Heart Association Task Force on Practice Guidelines. *J Am Coll Cardiol* 2013;62:e147-239.
4. Chaudhry SI, Mattera JA, Curtis JP, et al. Telemonitoring in patients with heart failure. *N Engl J Med* 2010;363:2301-9.
5. HeartLogic Heart Failure Diagnostic – Electrophysiology – Boston Scientific. (s. f.). [www.bostonscientific.com](http://www.bostonscientific.com). <https://www.bostonscientific.com/en-US/medical-specialties/electrophysiology/heartlogic-heart-failure-diagnostic.html>
6. HeartLogic Heart Failure Diagnostic – Electrophysiology – Boston Scientific. (s. f.). [www.bostonscientific.com](http://www.bostonscientific.com). <https://www.bostonscientific.com/en-US/medical-specialties/electrophysiology/heartlogic-heart-failure-diagnostic.html>
7. Quiénes somos. (s. f.). [www.bostonscientific.com](http://www.bostonscientific.com). <https://www.bostonscientific.com/es-MX/acerca-de/quienes-somos.html>
8. Vital signs (Body temperature, pulse rate, respiration rate, blood pressure) - Health Encyclopedia - University of Rochester Medical Center. (s. f.). <https://www.urmc.rochester.edu/encyclopedia/content.aspx?ContentTypeID=85&ContentID=P03963>
9. James, P. A., Oparil, S., Carter, B. L., Cushman, W. C., Dennison-Himmelfarb, C., Handler, J., ... & Smith Jr, S. C. (2014). 2014 evidence-based guideline for the management of high blood pressure in adults: report from the panel members appointed to the Eighth Joint National Committee (JNC 8). *JAMA*, 311(5), 507-520.
10. Benhayon, D. (2014). Impedancia transtorácica como herramienta para el manejo del paciente con insuficiencia cardíaca. *Insuficiencia cardíaca*, 9(1), 48-52. <https://www.redalyc.org/pdf/3219/321930870008.pdf>

11. Dr. Yitzar Bañuelos Millán, Cardiólogo en Los Mochis. (2019, 17 abril). Desfibrilador automático implantable | Dr. Yitzar Bañuelos Millán. Dr. Yitzar Bañuelos Millán.  
<https://cardiologialosmochis.com/portfolio/desfibrilador-automatico-implantable/>
12. Sanagustín, A. (2018, 20 marzo). Auscultación cardíaca : ruidos cardíacos, clics y chasquidos. Dr. Alberto Sanagustín. <https://www.albertosanagustin.com/2018/03/auscultacion-cardiaca-ruidos-cardiacos.html>
13. Moore, J. (2020). Robotic Process Automation: A Primer for the Digital Workforce. Wiley.
14. ¿Qué es el OCR? - Explicación del Reconocimiento óptico de Caracteres - AWS. (s. f.). Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/ocr/>
15. Greyrat, R. (2022, 5 julio). Python OpenCV – Operaciones morfológicas – Barcelona Geeks. [https://barcelonageeks.com/python-opencv-operaciones-morfologicas/#google\\_vignette](https://barcelonageeks.com/python-opencv-operaciones-morfologicas/#google_vignette)
16. Bradski, G., & Kaehler, A. (2008). Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly Media. Forsyth, D. A., & Ponce, J. (2012). Computer Vision: A Modern Approach. Prentice Hall.
17. Szeliski, R. (2010). Computer Vision: Algorithms and Applications. Springer.
18. Otsu, N. (1979). A Threshold Selection Method from Gray-Level Histograms. IEEE Transactions on Systems, Man, and Cybernetics, 9(1), 62-66.
19. González, R. C., Woods, R. E., & Eddins, S. L. (2009). Digital Image Processing Using MATLAB. Gatesmark Publishing.
20. Serra, J. (1982). Image analysis and mathematical morphology (Vol. 2). Academic press.
21. Soille, P. (2013). Morphological Image Analysis: Principles and Applications (2nd ed.). Springer.

## 10 Anexos

### 10.1 Partes del código: Métodos de Binarización

#### 1. \*Binarización invertida\*:

La binarización es un proceso en el que se convierte una imagen en escala de grises en una imagen binaria, donde los píxeles se clasifican como blanco o negro según un umbral determinado. En la binarización invertida, los píxeles que están por encima del umbral se convierten en negro y los que están por debajo se convierten en blanco. Esto es útil cuando queremos resaltar ciertos elementos en una imagen que son más oscuros que el fondo [17 y 18].

#### 2. \*Método de umbralización automática de Otsu\*:

La umbralización automática de Otsu es un método que determina automáticamente el umbral óptimo de binarización de una imagen en escala de grises. Este método se basa en el histograma de intensidad de la imagen y busca minimizar la varianza intra-clase, es decir, busca un umbral que maximice la separación entre las clases de píxeles (fondo y objeto) en la imagen. En otras palabras, busca un umbral que minimice la dispersión de los píxeles dentro de las clases y maximice la diferencia entre las clases [19].

En el código, la binarización invertida se logra mediante el parámetro `cv2.THRESH_BINARY_INV` en la función `cv2.threshold()`, y el método de umbralización automática de Otsu se activa con el parámetro `cv2.THRESH_OTSU`. Juntos, estos parámetros permiten convertir la imagen en una imagen binaria invertida donde los píxeles que representan líneas punteadas están en blanco y el fondo está en negro.

## 10.2 Partes del código: Operaciones morfológicas

A fin de esclarecer la aplicación de las operaciones morfológicas en los dos códigos iniciales, es necesario comprender algunas definiciones clave.

### 10.2.1 Erosión

La erosión de una imagen se realiza mediante una operación de convolución con un elemento estructurante (kernel) que recorre toda la imagen. En cada posición del kernel, se calcula el mínimo de los píxeles correspondientes en la imagen original. Esto tiene el efecto de "erosionar" los bordes de los objetos en la imagen, reduciendo su tamaño. [20 y 21]

Matemáticamente, la erosión de una imagen  $A$  con un kernel  $B$  se calcula como:

$$A \ominus B = \{z \mid (B)_z \subseteq A\} \quad (1)$$

Donde:

$z$  representa las coordenadas relativas del kernel o elemento estructurante en relación con el píxel central en el que se está realizando la operación.

Estas coordenadas se utilizan para acceder a los píxeles vecinos y realizar cálculos o comparaciones en función de la operación morfológica específica que se esté realizando.

$(B)_z$  representa la traslación de  $B$  por  $z$  y  $\subseteq$  denota la operación de intersección.

### 10.2.2 Dilatación

La dilatación de una imagen se realiza de manera similar a la erosión, pero en lugar de calcular el mínimo de los píxeles, se calcula el máximo. Esto tiene el efecto de "expandir" los objetos en la imagen, aumentando su tamaño y rellenando pequeños huecos o discontinuidades. [20 y 21]

Matemáticamente, la dilatación de una imagen  $A$  con un kernel  $B$  se calcula como:

$$A \oplus B = \{z \mid (B)_z \cap A \neq \emptyset\} \quad (1)$$

Donde  $(B)_z$  representa la traslación de  $B$  por  $z$  y  $\cap$  denota la operación de intersección.

### 10.2.3 Apertura

La apertura es el resultado de una erosión seguida de una dilatación. Es útil para remover ruido en las imágenes. [22]

#### 10.2.4 Close

En contraste, la clausura es una dilatación seguida de una erosión. Es buena para cerrar pequeños agujeros o puntos negros en los objetos en primer plano.[16]

Una vez que hemos adquirido una comprensión de las operaciones morfológicas, procedamos al análisis del código. En la ilustración correspondiente, se observa que el primer paso consiste en convertir la imagen a escala de grises, esto se realiza con el propósito de facilitar la siguiente conversión a formato binario. La imagen se convierte a binario, ya que este formato es requisito previo para la aplicación de operaciones morfológicas.

Posteriormente, se implementa la operación morfológica de apertura, la cual requiere la especificación de un kernel. Este kernel, conceptualmente similar a una brocha, se desplaza de izquierda a derecha y de arriba a abajo sobre la imagen, aplicando la transformación o efecto deseado. Es esencial entender que el kernel actúa como un filtro que modifica la estructura de la imagen según el tipo de operación morfológica que se busca realizar [16].

Se aplico esta operación morfológica debido a que mi imagen original tenía un fondo con líneas negras punteadas que causaban ruido, por lo que esta operación logro eliminar ese ruido de la imagen.

```
29 #Convertir a escala de grises
30 gray= cv2.cvtColor(crop,cv2.COLOR_BGR2GRAY)
31
32 #Filtro para detectar lineas punteadas, se utiliza la función "cv2.threshold" para realizar una binarización invertida de la imagen en
33 #escala de grises, donde las regiones que contienen lineas punteadas son representadas por píxeles blancos y el fondo e negro.
34 _, binaria = cv2.threshold(gray, 0, 255, cv2.THRESH_BINARY_INV + cv2.THRESH_OTSU) #
35
36 #Operación morfológica OPEN para eliminar las lineas punteadas, es una combinación de erosión seguida de dilatación.
37 #Ayuda a eliminar pequeños objetos y ruido de fondo mientras mantiene los objetos más grandes.
38 kernel= np.ones((3,3),np.uint8)
39 imagen_procesada= cv2.morphologyEx(binaria, cv2.MORPH_OPEN, kernel)
40
41 #Combinar la imagen original con la imagen procesada para conservar solo la linea
42 #Preservar solo las regiones de la imagen original donde la máscara tiene píxeles activos(distintos de cero)
43 imagen_final = cv2.bitwise_and(gray, gray, mask=imagen_procesada)
44
45
```

*Ilustración 4. Operación morfológica de apertura aplicada en el código.*

Tras el procesamiento inicial de la imagen, se identificaron algunos espacios vacíos en el área de interés, lo que motivó la necesidad de aplicar otra operación morfológica, específicamente la de cierre. Para llevar a cabo este paso, inicialmente, se procedió a definir el kernel mediante un proceso de prueba y error. Una vez establecido, se aplicó la operación de cierre con el objetivo de eliminar pequeños agujeros en los objetos del primer plano.

Posteriormente, se llevaron a cabo procedimientos adicionales, entre los cuales se incluyó la identificación de contornos. Este paso permitió rellenar los contornos con blanco, homogeneizando así la imagen con la versión original.

```

53 #Operacion morfologia de close, es una combinación de dilatación seguida de erosión.
54 #Sirve para rellenar huecos en los contornos y unir objetos que están separados por espacios pequeños
55 kernel = np.ones((7,7),np.uint8)
56 imagen_cerrada= cv2.morphologyEx(imagen_final, cv2.MORPH_CLOSE, kernel)
57 #cv2.imshow('Imagen cerrada',imagen_cerrada)
58 #cv2.waitKey(0)
59 #cv2.destroyAllWindows()
60
61 #Encontrar contornos, excluye los contornos dentro de contornos y elimina los puntos redundantes y comprime el contorno
62 contornos, _=cv2.findContours(imagen_cerrada, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
63
64 #Rellenar de blanco
65 cv2.drawContours(imagen_cerrada,contornos,-1,(255,255,255), thickness=cv2.FILLED) #Se rellenan de color blanco los contornos
66 imagen_invertida = cv2.bitwise_not(imagen_cerrada) #Contornos estén en un fondo blanco
67

```

*Ilustración 5. Operación morfológica de cierre aplicada en el código.*