



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

---

**FACULTAD DE INGENIERÍA**

**Desarrollo del Módulo M para  
SYSCARDIO, Sistema Inteligente  
para el diagnóstico médico para  
enfermedades Cardiovasculares**

**TESINA**

Que para obtener el título de  
**Ingeniero en Computación**

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

Joshua Curiel Reyes

**DIRECTORA DE TESINA**

Dra. Jimena Olveres Montiel



Ciudad Universitaria, Cd. Mx., 2023

## Agradecimientos

---

La culminación de este trabajo representa la suma de todos los esfuerzos a lo largo de esta única y extraordinaria etapa de mi vida que sin duda no hubiera logrado concluir sin el apoyo de todas las personas que en mi camino me apoyaron, brindándome su conocimiento, su expertise y fomentando en mí una formación no únicamente ingenieril sino también una formación personal y espiritual.

Mi más grande agradecimiento siempre será hacia Dios quien es el más grande maestro dándome siempre la sabiduría, la inteligencia, la resiliencia y la fuerza para continuar en mi camino.

Cómo un auto-agradecimiento dedico estas palabras a ese joven Joshua Curiel quien emprendió este camino siempre con la firme convicción de lograr sus objetivos siendo resiliente ante las adversidades y los retos, sintiéndose cada día más orgulloso de sí mismo y dibujando una sonrisa no únicamente en su rostro sino en el de todo aquel que coincidía en su camino.

Este logro es para mi familia quienes son el pilar más grande que sostiene mi existir, a mis padres que son la razón por la que yo esté aquí y que siempre han tenido las palabras correctas que me han impulsado a seguir adelante, siempre con el amor y el apoyo para recordarme que soy capaz de lograr todo lo que me proponga. A mis abuelitos que con sabiduría y una gran sonrisa siempre le han dado paz y tranquilidad a mi corazón y a mi pensamiento.

Con todo el amor de mi alma, con mi entera gratitud, admiración y con nostalgia agradezco y le dedico este logro a mi hermano Abraham Curiel Reyes que sin duda para mí siempre será el mejor hombre y ser humano que pude haber conocido y que ahora es el ángel que me acompaña en todo momento. Siempre serás cómo un padre para mí hermano, tú me has cuidado, enseñado, levantado cuando he caído, tú que me ayudaste cuando estaba exhausto y no podía más, hermano tú siempre has sido y serás la inspiración que me ha llevado y me seguirá llevando a lograrlo todo en esta vida. Esto es para tí hasta dónde sea que te encuentres Abraham.

Para mi hermoso pequeñito Máximo Curiel Abad, que Dios te cuide y te dé salud y vida y a mí para cuidarte, protegerte y guiarte tal cómo tu papá lo hizo conmigo. Mi pequeño debes mirar hacia adelante siempre, llevando en tu pensamiento y en tu corazón a quienes ya no están, siempre haciendo frente a las adversidades de esta vida cómo un guerrero, con resiliencia, fe y fortaleza. Eres mi fortaleza Max, en tí aún puedo ver lo hermoso de esta vida, y aunque siempre tendrás un sólo padre que es Abraham, parte de mi misión en esta vida es guiarte cómo un padre. Sólo me queda decirte vive, sonríe, ama y agradece a Dios por un día más.

A mi amada pachis Jenny, le agradezco a Dios por haberte puesto a mi lado en el momento indicado, gracias por tener la asombrosa capacidad de tocar mi corazón y devolverle ese brillo, ese destello, eres un hermoso ser que volvió a dibujar esa sonrisa en mi rostro, quédate a mi lado y tomados de la mano y con Dios en nuestra vida lograremos todo lo que nos propongamos.

Agradezco a mi casa, mi alma mater, la Universidad Nacional Autónoma de México que desde mi ingreso al Colegio de Ciencias y Humanidades Plantel Oriente hasta el último día de mis clases en mi amada Facultad de Ingeniería, me abrió las puertas hacia esta extraordinaria aventura llena de altibajos pero que siempre me dejaron una valiosa enseñanza que a dónde sea que vaya me permiten poner en alto el nombre mi universidad.

Agradezco a mis amigos Andrés, José Luis, Orlando, Jorge, Sergio, Daniel, Antonio, Osmar, Daniela, Irma, Luisa, gracias por hacer está travesía tan alegré, divertida, y sobre todo por mostrarme el verdadero significado del compañerismo, la amistad y la hermandad.

Agradezco a mis profesores, que siempre me brindaron su apoyo incondicional. Cómo menciones especiales en mi etapa cómo estudiante de CCH me gustaría agradecer a la profesora Daría Razo Balcázar y al profesor Gerardo Francisco Gómez por ser ambos una gran influencia para mí en esa etapa de mi vida.

Agradezco a la Dra. Jimena Olveres y al Dr. Boris Escalante, por brindarme la oportunidad de colaborar en uno de los mejores laboratorios de investigación a nivel internacional (LaPI), por sus enseñanzas cómo profesores y por todo el apoyo, sus observaciones y correcciones que hicieron posible la realización de esta tesina. A mi gran amigo y mentor el Dr. Fabian Torres Robles quien siempre con un extraordinario positivismo y paciencia me enseñó y transmitió su conocimiento siendo una parte fundamental para la realización de está tesina, eres brillante y una excelente persona amigo.

El presente trabajo agradece el apoyo de los programas de apoyo: UNAM PAPIIT IV100420 y SECTEI/202/2019.

## Índice general

1. Introducción.....	4
1.1. Justificación.....	6
1.2. Objetivo .....	8
2. Marco Teórico.....	9
2.1. Ultrasonido para ecocardiografía.....	10
2.2. M mode.....	13
2.3. Tapse mode.....	17
2.4. VTK .....	19
2.5. Qt Framework.....	22
3. Metodología.....	24
3.1. Creación de módulo M mode.....	25
4. Pruebas y Resultados.....	35
5. Conclusiones.....	41
6. Referencias.....	42
7. Anexo: Manual de Usuario sistema SYSCARDIO .....	45

# CAPÍTULO 1

## Introducción.

---

El constante desarrollo y evolución en el área tecnológica trae consigo grandes beneficios en diversas áreas del conocimiento, además de ser un pilar para mejorar la condición humana. La necesidad de obtener una respuesta oportuna y eficiente en los sistemas que poseen las diversas capacidades para brindar apoyo y soporte en la realización de una o más tareas específicas son la base introductoria al vínculo que se origina entre el área de Ingeniería en Computación y el área Médica.

Un análisis eficiente y oportuno contribuye de manera significativa para la síntesis de un diagnóstico el cual conduzca a la pronta atención del paciente, reduciendo los tiempos de respuesta ante una anomalía cardíaca detectada con ayuda del sistema.

La hipertensión arterial pulmonar (HAP) es una condición compleja y de mal pronóstico. Es definida como un aumento en la presión arterial pulmonar media  $> 25$  mmHg, determinada por cateterismo cardíaco derecho y con presión de enclavamiento pulmonar  $< 15$  mmHg. La prevalencia a nivel mundial es de 15 casos por millón de población. (Ordoñez & Montes , 2015)(Sandoval Zarate, 2006).

La prevalencia estimada de HAP a nivel mundial es de 30-50 casos/millón de población adulta. En México, así como en América Latina, no existen datos específicos sobre la prevalencia (Ordoñez & Montes, 2015).

Es de suma importancia que el médico de primer contacto pueda conocer dicha patología, así como de la historia natural de la enfermedad y, de esa manera, iniciar los algoritmos diagnósticos comentados previamente para clasificar adecuadamente al paciente y derivar de manera oportuna a los centros especializados en HAP (Ordoñez & Montes, 2015).

El Laboratorio Avanzado de Procesamiento de Imágenes de la Facultad de Ingeniería de la UNAM en conjunto con la Robert Gordon University of Aberdeen Escocia llevan a cabo la realización de un proyecto el cual lleva por nombre

SYSCARDIO: Sistema Inteligente para el diagnóstico médico para enfermedades Cardiovasculares, el cual es un sistema que asiste en el diagnóstico de enfermedades cardiovasculares a los profesionales de la salud. SYSCARDIO provee las herramientas que permiten la visualización, manejo y procesamiento de imágenes con formatos propios del área médica.

Cómo parte del desarrollo del sistema es necesario tomar las consideraciones pertinentes partiendo siempre del enfoque propuesto por el especialista médico, que en este caso es el desarrollo de un sistema que permita el análisis, manejo y manipulación de imágenes de la región cardíaca a la vista de cuatro cámaras. Dicho lo anterior esto ha sido posible mediante la implementación de métodos de visión computacional e inteligencia artificial, que ofrecen un panorama más amplio y son un claro ejemplo de la contribución del avance tecnológico en la resolución de problemas en beneficio de la condición humana.

Python es un lenguaje de programación de alto nivel interpretado, orientado a objetos y con semántica dinámica. Sus estructuras de datos integradas de alto nivel, combinadas con la tipificación dinámica y el enlace dinámico, lo hacen muy atractivo para el desarrollo rápido de aplicaciones ( Python T.M., 2022).

Visualization ToolKit (VTK) es un sistema de software de código abierto disponible gratuitamente para gráficos por computadora en 3D, procesamiento de imágenes y visualización utilizado por miles de investigadores y desarrolladores en todo el mundo. VTK consta de una biblioteca de clases de C++ y varias capas de interfaz interpretadas, incluidas Tcl/Tk, Java y Python. (Gasser, 2010)(Marcus D. H, 2015).

## **CAPÍTULO 1.1.**

### **Justificación**

---

La hipertensión pulmonar es un síndrome clínico grave, multifactorial y multidisciplinario, con gran perjuicio para la calidad de vida de los pacientes y con alta morbimortalidad (Madrid Dour, 2015).

México registra la prevalencia más alta de hipertensión arterial en todo el mundo, de acuerdo con la Encuesta Nacional de Salud y Nutrición 2012, y ello se debe a la falta de un diagnóstico oportuno y al desconocimiento de los factores de riesgo que pueden detonar que se eleve la presión arterial (IMSS, 2017).

Uno de cada tres mexicanos mayores de edad padece hipertensión arterial, enfermedad crónica degenerativa cardiovascular que registra 7 millones de casos y provoca más de 50 mil muertes al año (IMSS, 2017).

El ventrículo derecho juega un rol importante en la morbilidad y mortalidad de los pacientes que se presentan con signos y síntomas de enfermedad cardiopulmonar. Sin embargo, la evaluación sistemática de la función cardíaca derecha no es uniformemente llevada a cabo. Esto es en parte debido a la enorme atención dada a la evaluación de las cámaras izquierdas, a la falta de familiaridad con las técnicas ultrasonográficas que pueden ser usadas en la obtención de imágenes del corazón derecho, y a la escasez de estudios en ultrasonido que brinden valores normales de referencia sobre la función y el tamaño del corazón derecho (Lang RM et al., 2015)(Bouzas B., 2009).

La excursión sistólica del anillo tricúspide en dirección al ápice denota ser una buena técnica para evaluación del VD (Madrid Dour, 2015).

La mayoría de los sistemas de análisis de imágenes médicas comerciales, especialmente los desarrollados por las compañías proveedoras de equipo de imagenología médica, incluyen utilerías destinadas a calcular parámetros clínicos

del ventrículo izquierdo, sin embargo, ninguno incluye la posibilidad de realizar estudios similares en otras cavidades cardiacas.

Dicho lo anterior el sistema de monitoreo cardiaco SYSCARDIO cómo punto de partida para el desarrollo de esta tesina enfocada en el análisis, construcción y prueba del módulo de M mode, busca resolver necesidades específicas descritas por los médicos especialistas, adaptándose a cada una de las etapas del proceso para la generación de un diagnóstico completo y eficiente.

La ventaja del modo M es su tasa de muestreo muy alta, lo que da como resultado una alta resolución de tiempo, de modo que incluso los movimientos muy rápidos pueden grabarse, mostrarse y medirse.

De acuerdo con una estudio descriptivo de pacientes adultos atendidos en un servicio de ecocardiografía que tenía por objetivo el determinar el valor de referencia del desplazamiento sistólico del anillo tricúspide (TAPSE) en una población de pacientes sin comorbilidades se llegó a la conclusión que en pacientes con un amplio rango de edad sin enfermedad cardiaca izquierda o derecha y sin hipertensión pulmonar, el valor del TAPSE osciló entre 22 mm y 23 mm, con un valor inferior de normalidad de 18 mm para la población evaluada (López Nilson MD et al. 2012).



## **CAPÍTULO 1.2.**

### **Objetivo**

---

Desarrollar, mejorar y probar un módulo que emulará la técnica de ecocardiografía Modo M, el cual se integra con el sistema de monitoreo cardiaco SYSCARDIO como parte del proceso que tiene como propósito la asistencia en el análisis cualitativo y cuantitativo del movimiento de las estructuras cardiacas provistas por el sistema.

## **CAPÍTULO 2.**

### **Marco Teórico**

---

Dada la naturaleza del problema que se aborda en esta tesina, el ultrasonido para ecocardiografía es el pilar para la obtención, visualización y manejo de la información de la estructura cardiaca, describiendo los modos y las vistas del ultrasonido. Es por medio de este que es posible la valoración del estado de salud de los pacientes.

Nuestra comprensión de los fundamentos físicos de la evaluación de la función cardíaca es cada vez más sofisticada. Los desarrollos recientes en ultrasonido cardíaco permiten la explotación de muchos de los conceptos físicos más nuevos con el equipo ecocardiográfico actual (D. Thomas & B. Popovic, 2006).

La hipertensión pulmonar se define como un grupo de enfermedades caracterizadas por el incremento progresivo de la resistencia vascular pulmonar (RVP), que da lugar a insuficiencia ventricular derecha y muerte prematura.

El enfoque en el desarrollo del sistema **SYSCARDIO** está orientado al ventrículo derecho de la estructura cardiaca debido a que la naturaleza anatómica del mismo dificulta el análisis en la progresión de la Hipertensión Arterial Pulmonar que trae consigo como consecuencia la postcarga del ventrículo derecho, resultando en insuficiencia cardiaca derecha. Ante la sospecha clínica de HAP, debe realizarse ecocardiograma y confirmar el diagnóstico sólo con cateterismo cardiaco derecho (Ordoñez & Montes, 2015).

La funcionalidad del módulo de modo M es una de las herramientas las cuales ofrece el sistema **SYSCARDIO** al profesional del área de la salud el cual debe tener los conocimientos suficientes para poder dar una interpretación correcta de los datos de salida de este módulo de **modo M**.

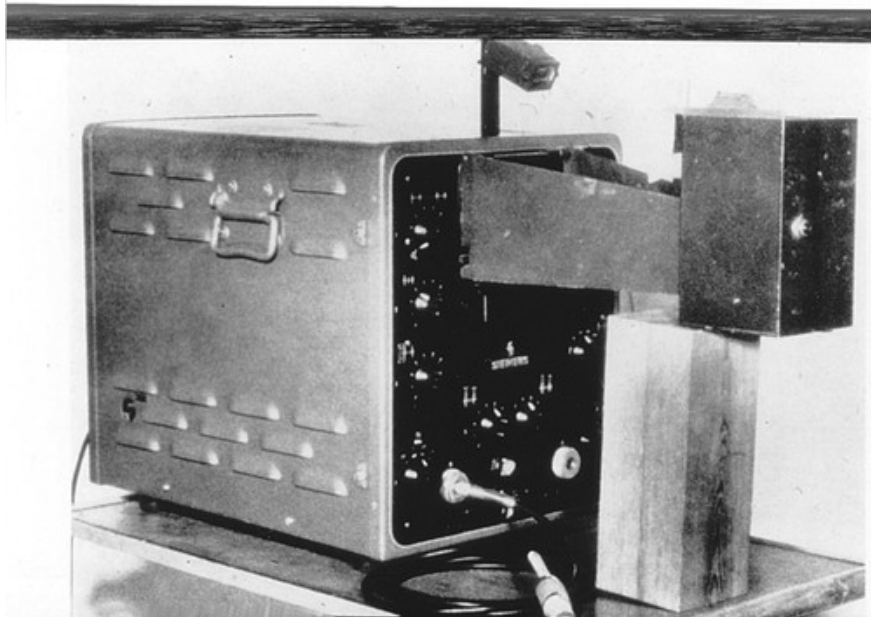
## CAPÍTULO 2.1

### Ultrasonido para ecocardiografía

---

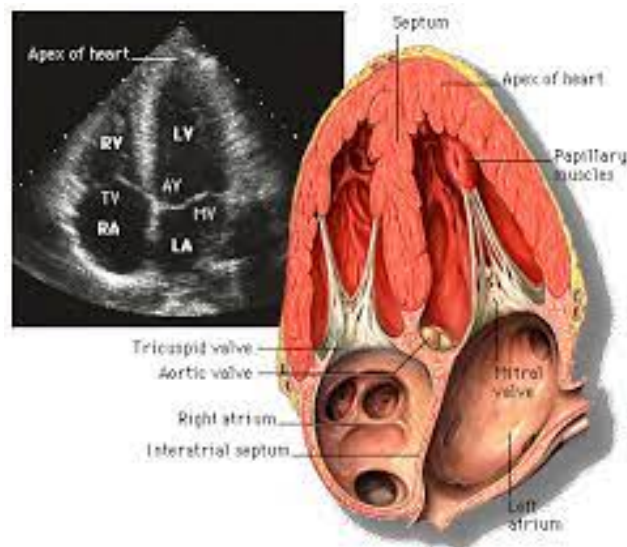
La ecocardiografía es el uso de ultrasonido para evaluar los componentes estructurales del corazón en una estrategia mínimamente invasiva. Aunque, antes de la invención de la ecocardiografía bidimensional de uso habitual en la actualidad, existía la ecocardiografía basada en el movimiento (modo M) (Omerovic Silvan, 2021).

La descripción original de la ecocardiografía en modo M en 1953, por parte de Inge Edler (1911–2001) y su amigo el físico Hellmuth Hertz, marcó el comienzo de una nueva técnica de diagnóstico no invasiva. Edler utilizó esta técnica principalmente para el estudio preoperatorio de la estenosis mitral y el diagnóstico de la insuficiencia mitral, cómo se muestra en la Figura 1 uno de los primeros equipos que utilizó Elder para registrar ecogramas en modo M (Singh and Goyal, 2007).



**Figura 1. Primeros equipos ecocardiográficos utilizados por Edler y Hertz para registrar ecogramas en modo M (Tomado de (H. Feigenbaum, 1996)).**

Un ecocardiograma verifica cómo las cámaras y válvulas del corazón están bombeando sangre a través de este. Un ecocardiograma usa electrodos para controlar el ritmo cardíaco y tecnología de ultrasonido para ver cómo se mueve la sangre a través del corazón. Un ecocardiograma puede ayudar al médico a diagnosticar enfermedades del corazón por medio de visualizaciones cómo la que se muestra en la Figura 2 representando el ecocardiograma Doppler(Libby, 2021).



**Figura 2. Ecocardiograma Doppler. (Tomado de (Ortego, O. A. (2020))**

De acuerdo con la visualización del plano ecocardiográfico del sistema **SYSCARDIO** es necesario abordar el plano que en ámbito ecocardiográfico se denomina “**Apical de cuatro cámaras**”, dónde el paciente en posición lateral izquierda, se coloca el transductor en el vértice. En esta vista, las cuatro cámaras del corazón se ven simultáneamente. El ventrículo izquierdo aparece como una elipse truncada, con visualización del tabique interventricular, el vértice y las paredes laterales. La pared lateral del ventrículo izquierdo se muestra en el lado derecho de la pantalla. La pared libre basal, media y apical del ventrículo derecho se ven en el lado izquierdo de la pantalla, con la banda moderadora conectando la pared libre con el tabique. En la parte inferior de la pantalla se encuentran las aurículas izquierda y derecha y el tabique interauricular (Mohamed A. A., 2010) (Ayuela Azcaratea et al., 2012).

Las venas pulmonares se ven en la parte inferior del sector de la imagen superior a la aurícula izquierda. La valva mitral anterior aparece medialmente y la valva posterior lateralmente. También se ven las valvas tricuspídea septal y lateral. La valva septal de la válvula tricúspide está desplazada apicalmente con respecto a la válvula mitral, las cuales se pueden apreciar en la Figura 3 representando el Apical de cuatro cámaras (Mohamed A. A., 2010).

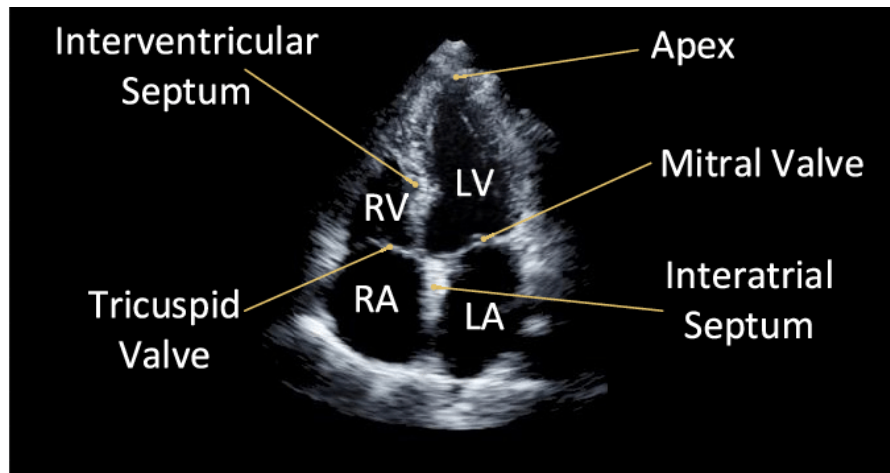


Figura 3. Apical de cuatro cámaras (Tomado de (Jafari, Mohammad & Girgis, 2019)).

Según la información que necesite el médico, es posible que se realice uno de varios tipos de ecocardiogramas. Cada tipo de ecocardiograma implica pocos riesgos, si es que los hay.

Hay varios modos de ultrasonido de imagen conocidos, el modo A, modo B, modo M, modo en tiempo real, Doppler de onda pulsada, Doppler de onda continua y Doppler color. En esta tesina se aborda el modo de ultrasonido “modo M” (Mohamed A. A., 2010) (Khandheria Bijoy K. M.D., 1994).

## CAPÍTULO 2.2

### Modo M

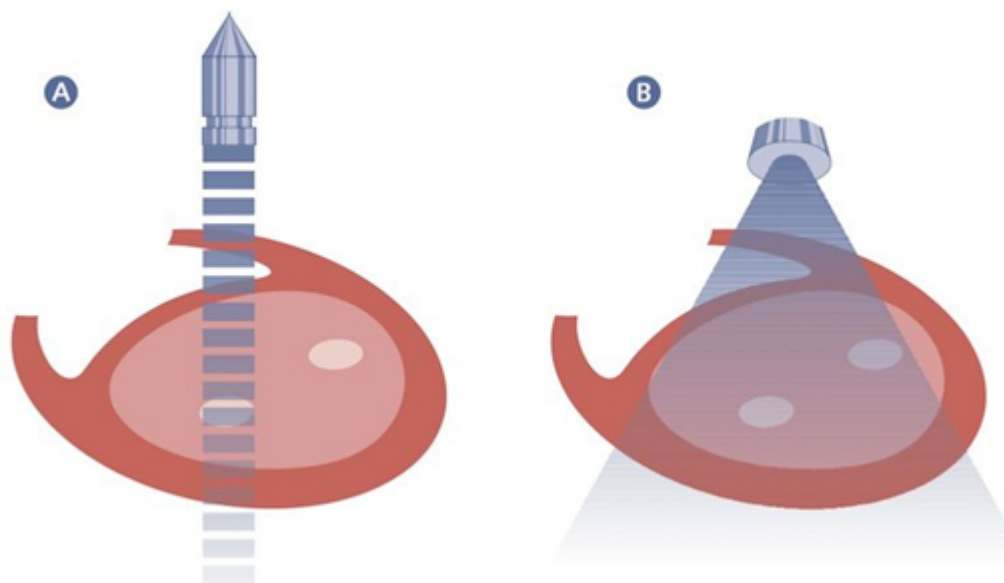
---

El **modo M** es la base de esta tesina, es el modo en ultrasonido para ecocardiografía para la visualización del movimiento en el tiempo, por ello es fundamental que se defina el concepto de esta modalidad y su papel dentro del desarrollo de este módulo del sistema **SYSCARDIO**.

El **modo M** era la modalidad de imagen preferida en los primeros días de la ecografía. El modo M se define como la visualización del movimiento en el tiempo de la onda de ultrasonido a lo largo de una línea de ultrasonido elegida. Proporciona una vista mono dimensional del corazón (Binder, 2020).

Para poder dar forma y funcionalidad a este módulo como parte del sistema **SYSCARDIO** es necesario conocer la definición y cada una de las implicaciones del modo M, que siguiendo este análisis la Sociedad Española de Imagen Cardíaca nos define lo siguiente:

**El modo M consiste en la emisión de un haz de ultrasonidos único que atraviesa las distintas estructuras cardíacas, las cuales generan ecos reflejados, cuyo movimiento en el espacio se registra en papel o en pantalla de manera continua, obteniendo las características de movilidad de las diferentes zonas del corazón atravesadas por el haz de ultrasonidos (G. F. M. A. García Fernández & Zamorano, 2020).**



**Figura 4. A) Se muestra el haz de rayos que analiza el corazón con la técnica en modo M. Como puede observarse se estudia un sector muy limitado del miocardio izquierdo; B) Se muestra el sector que analiza el corazón con ecocardiografía bidimensional. Como puede observarse permite analizar sectores amplios del corazón (Tomado de G. F. M. A. García Fernández & Zamorano, 2020)).**

Para el registro del ecocardiograma en modo M el transductor de ultrasonidos se sitúa en la pared torácica sobre el tercero, cuarto o quinto espacio intercostal. El cambio de incidencia del haz permite registrar las diferentes estructuras cardíacas (tal cómo se muestra en la Figura 4), cuyas características ecocardiográficas se describen a continuación:

**Raíz aórtica:** Se visualiza como una estructura formada por un eco anterior, que corresponde a la pared anterior aórtica, y un eco posterior, que corresponde a la pared posterior; entre estas dos estructuras se registra el movimiento de los velos sigmoideos aórticos. Durante la diástole el eco permanece situado en medio de la luz del vaso; durante la sístole se registra el movimiento de apertura de los velos sigmoideos aórticos. Por detrás de la raíz aórtica se localiza la cavidad auricular izquierda y por delante el ventrículo derecho (G. F. M. A. García Fernández & Zamorano, 2020).

**Válvula mitral:** El velo anterior mitral se caracteriza por presentar, en protodiástole, un rápido movimiento de apertura (punto E), para luego iniciar un cierre mesodiastólico; la contracción auricular producirá una reapertura de la válvula

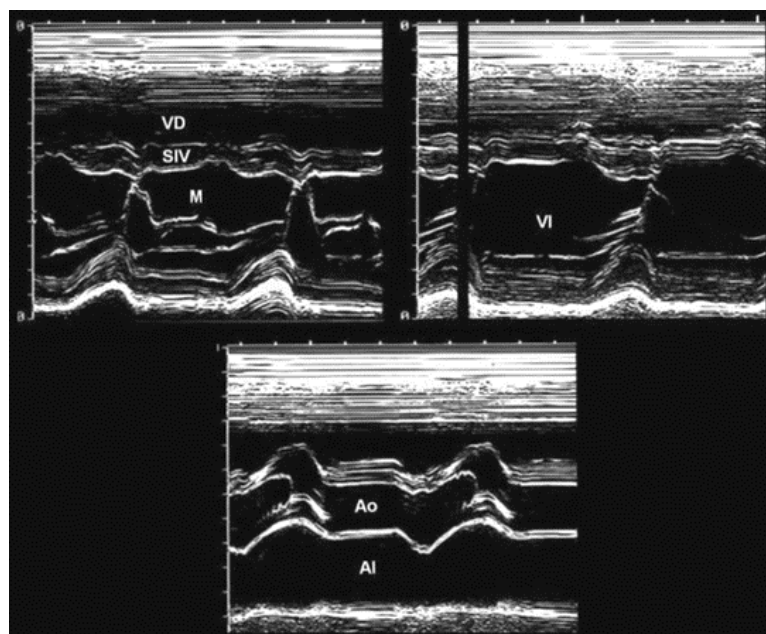
(punto A), tras la cual aparece el cierre por la oposición de los velos anterior y posterior. El velo posterior presenta, durante la diástole, un movimiento especular opuesto al del velo anterior, que desaparece en caso de afectación reumática mitral (G. F. M. A. García Fernández & Zamorano, 2020).

**Válvula tricúspide:** Dirigiendo el transductor hacia la válvula tricúspide se registra su movimiento de apertura y cierre de características similares, al de la válvula mitral (G. F. M. A. García Fernández & Zamorano, 2020).

**Ventrículo izquierdo:** La cavidad ventricular aparece en la ecocardiografía en modo M entre el tabique interventricular y la pared posterior del ventrículo izquierdo. Durante la sístole se registra el engrosamiento del miocardio septal y de la pared posterior del ventrículo izquierdo, con disminución del diámetro ventricular (diámetro sistólico). Durante el llenado ventricular se produce un aumento progresivo del tamaño ventricular hasta alcanzar el diámetro diastólico (G. F. M. A. García Fernández & Zamorano, 2020).

A partir de estas dos mediciones mencionadas en el párrafo anterior pueden inferirse patrones básicos de la dinámica cardíaca, como son los volúmenes telediastólico y telesistólico, la fracción de eyección y la fracción de acortamiento (G. F. M. A. García Fernández & Zamorano, 2020).

Las características ecocardiográficas descritas anteriormente pueden ser visualizadas en la Figura 5.





**Figura 5.** En la imagen se pueden apreciar los diferentes cortes haciendo uso del Modo M. En la parte superior izquierda el movimiento de la válvula mitral (**M**), anterior a esta se puede visualizar el ventrículo izquierdo (**VD**), e inmediatamente posterior el septo interventricular (**SIV**); en la parte superior derecha dirigiendo el transductor con una angulación más inferior, se registra el ventrículo izquierdo (**VI**) enmarcado por el tabique y la pared posterior; por último en la parte inferior central en la raíz aórtica (**Ao**) se registran las paredes aórticas anterior y posterior y, dentro de la luz aórtica, el movimiento de las valvas sigmoideas aórticas (Tomado de G. F. M. A. García Fernández & Zamorano, 2020).

El **modo M** proporciona una evaluación rápida de múltiples valores de espesor de pared, diámetros de cámara, fracción de eyección, gravedad de la insuficiencia aórtica y más. El modo M es una sola línea de visión, por lo que no tiene en cuenta los cambios en el corazón en otras áreas además de su único haz de ultrasonido. A pesar de estas limitaciones, el Modo M es preciso en la mayoría de los casos y es una medida estándar en TTE (JLS Interactive, LLC. 2010).

Una vez descritos los principios básicos del **modo M** vamos a abordar la forma en cómo se puede utilizar este modo para la construcción de la imagen del modo M y dónde se debe utilizar.

El **modo M** se puede realizar desde una vista de eje largo o corto. Independientemente de la vista que utilice, se debe asegurar el alinear las estructuras de forma perpendicular al modo M. Esto se hace ajustando el plano de la imagen 2D y la línea del modo M. También es importante investigar la raíz aórtica mediante el Modo M a nivel de las cúspides aórticas (Binder, B. T. 2020)

## CAPÍTULO 2.3

### TAPSE o Movimiento Anular Tricúspideo (MAT).

---

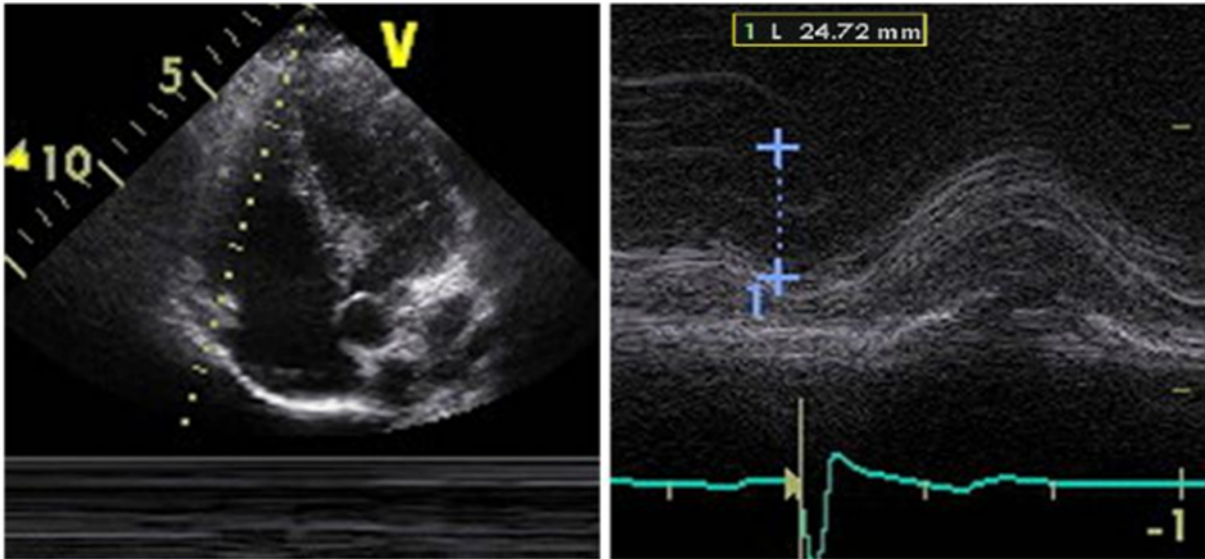
El TAPSE o MAT es un método para medir la distancia de la excursión sistólica del segmento anular del VD a lo largo de su plano longitudinal, a partir de una ventana apical de 4 cámaras. El TAPSE o MAT representa la función longitudinal del VD de la misma forma que la excursión sistólica del anillo mitral evaluada por Doppler tisular lo hace en el VI (Fuentes Molina Edgar Dr. & Lowenstein Jorge Dr., 2010).

La excursión sistólica del plano anular tricuspídeo (TAPSE) se ha convertido en una herramienta popular para evaluar la función sistólica del ventrículo derecho (VD) debido a su facilidad de aplicación. La TAPSE con ecocardiografía transesofágica (ETE) está limitada por la alineación con la pared lateral del VD (Makhija, 2019)(R. Forfia et al., 2006).

La propensión a usar TAPSE para evaluar la función sistólica del VD se debe a su fácil aplicación y alta reproducibilidad. TAPSE ahora está incluido en las pautas de la American Society of Echocardiograph (ASE) para la evaluación ecocardiográfica del corazón derecho en el adulto (Makhija, 2019).

A partir de la construcción de una imagen en modo M podemos obtener la medición de la excursión sistólica del plano anular tricuspídeo del anillo lateral tricuspídeo (TAPSE).

El TAPSE es usualmente obtenido colocando el cursor del modo M a través del anillo tricuspídeo, y midiendo la cantidad de movimiento longitudinal del anillo en el pico de la sístole (ver Figura 6).



**Figura 6. Medición de la excursión sistólica del anillo tricuspídeo (TAPSE) (Tomado de Dr. Edgar Fuentes Molina, Dr. Jorge Lowenstein . (2010)).**

Cómo todo método el Modo Tapse resalta ventajas y desventajas en su uso, para la obtención de información dependiente de la previa y correcta construcción de la imagen en Modo M, dichas ventajas y desventajas se describen a continuación.

- **Ventajas:** El TAPSE es simple, menos dependiente de una calidad óptima de imagen, y reproducible, y no requiere equipo sofisticado o un análisis prolongado de la imagen (Fuentes Molina Edgar Dr. & Lowenstein Jorge Dr., 2010).
- **Desventajas:** El TAPSE asume que el desplazamiento de un único segmento representa la función de una compleja estructura tridimensional. Además, es ángulo dependiente, y no hay estudios de validación a gran escala. Finalmente, el TAPSE puede ser carga dependiente (Fuentes Molina Edgar Dr. & Lowenstein Jorge Dr., 2010).

**NOTA:** Recomendaciones: El TAPSE debería de ser usado rutinariamente como un método simple para estimar la función del VD, con un valor de referencia inferior como corte para deterioro de la función sistólica del VD de **16 mm** (Fuentes Molina Edgar Dr. & Lowenstein Jorge Dr., 2010).

## CAPÍTULO 2.4

### VTK.

---

En este capítulo se aborda una de las librerías fundamentales para la creación, uso, manejo, y mejora del módulo de Modo M, ya que sienta un kit de herramientas de visualización nos ofrece una amplia gama de componentes y funciones para el procesamiento, tratamiento y modificación de las imágenes, secuencias y videos de ultrasonido.

Visualization Toolkit (VTK) es un sistema de software de código abierto y disponible gratuitamente para gráficos 3D por computadora, modelado, procesamiento de imágenes, representación de volumen, visualización científica y trazado 2D. Admite una amplia variedad de algoritmos de visualización y técnicas de modelado avanzadas, y aprovecha el procesamiento paralelo de memoria distribuida y en subprocesos para lograr velocidad y escalabilidad, respectivamente (Schroeder Will et al., 2006).

VTK dispone de tres versiones principales para su descarga e implementación desde su sitio oficial. La versión 7.1.1 es la versión más antigua disponible, por otro lado teniendo como opción la versión intermedia la 8.2.0, y finalmente la versión 9.2.2 como su "last release", cada una de estas con su correspondiente documentación disponible para descargar en conjunto con el archivo instalable de acuerdo con el sistema operativo en el que se desee correr.

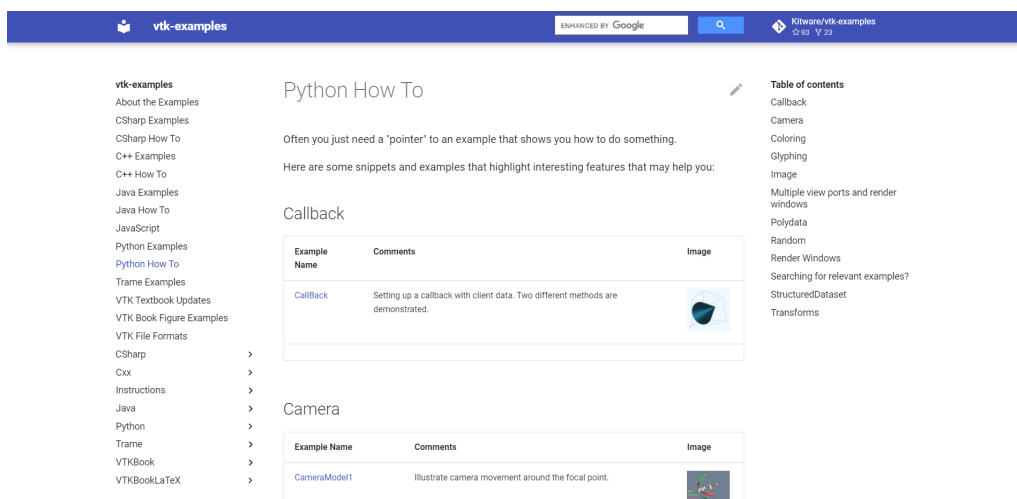
Los wrappers de Python para VTK se producen algorítmicamente a partir de los archivos de encabezado de VTK, y usar VTK a través de Python es muy parecido a usar VTK a través de C++. Casi toda la misma funcionalidad está disponible, con tipos de Python convertidos automáticamente a tipos de C++ y viceversa.

Kitware provee de una documentación oficial de VTK, además de la descripción detallada de una serie de ejemplos en la implementación de diversos lenguajes de programación, como lo son Java, JavaScript, C++, C#, LaTeX y por supuesto Python dando pie a la interoperabilidad entre estos lenguajes, los recursos

y las herramientas que brinda VTK. Una vista previa de dicha documentación se muestra en las Figuras 7 y 8.

La documentación correspondiente a la implementación de VTK en Python abarca algunos de los siguientes contenidos:

- Callback
- Camera
- Coloring
- Glyphing
- Image
- Multiple view ports and render windows
- Polydata
- Random
- Render Windows
- StructuredDataset
- Transforms



**Figura 7. Documentación oficial de ejemplos e implementaciones de Python en otros lenguajes (Tomado de (Schroeder Will et al., 2006)).**

```

#!/usr/bin/env python

# This simple example shows how to do basic rendering and pipeline
# creation.

# noinspection PyUnresolvedReferences
import vtkmodules.vtkInteractionStyle
# noinspection PyUnresolvedReferences
import vtkmodules.vtkRenderingOpenGL2
from vtkmodules.vtkCommonColor import vtkNamedColors
from vtkmodules.vtkFiltersSources import vtkCylinderSource
from vtkmodules.vtkRenderingCore import (
    vtkActor,
    vtkPolyDataMapper,
    vtkRenderWindow,
    vtkRenderWindowInteractor,
    vtkRenderer
)

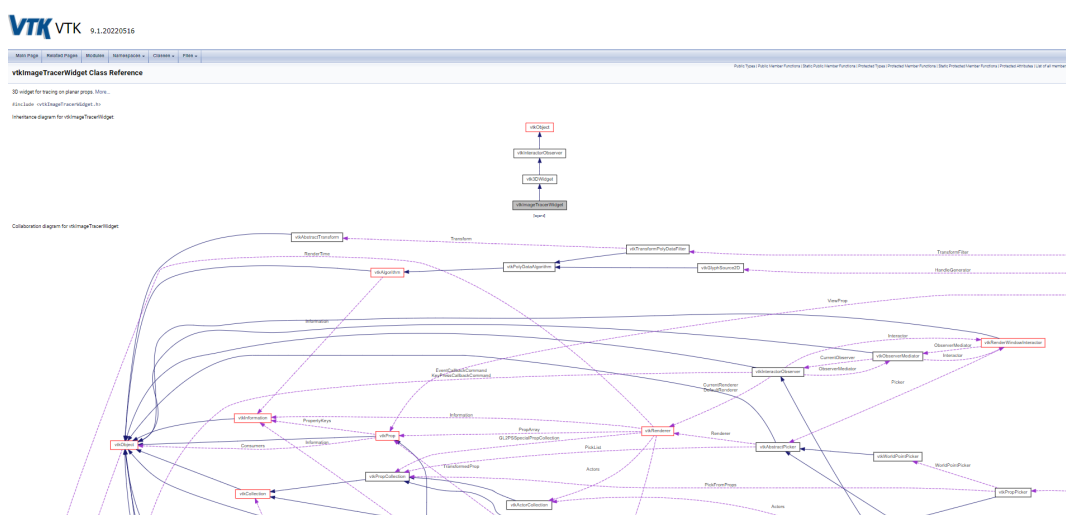
def main():
    colors = vtkNamedColors()
    # Set the background color.
    bkg = map(lambda x: x / 255.0, [26, 51, 102, 255])
    colors.SetColor("BkgColor", *bkg)

    # This creates a polygonal cylinder model with eight circumferential
    # facets.
    cylinder = vtkCylinderSource()
    cylinder.SetResolution(8)

    # The mapper is responsible for pushing the geometry into the graphics
    # library. It may also do color mapping, if scalars or other
    # attributes are defined.
    cylinderMapper = vtkPolyDataMapper()
  
```

**Figura 8. Documentación de ejemplos que provee Kitware para la implementación en Python de VTK (Tomado de (Schroeder Will et al., 2006)).**

La documentación oficial de VTK nos provee de una descripción detallada de cada una de las clases de VTK, además de ello nos ofrece un diagrama mostrando la jerarquía de dónde hereda la clase que estamos consultando, tal cómo se muestra en la Figura 9.



**Figura 9. vtkImageTracerWidget Class Reference (Tomado de (VTK Wiki, s.f.)).**

## CAPÍTULO 2.5

### Qt Framework

---

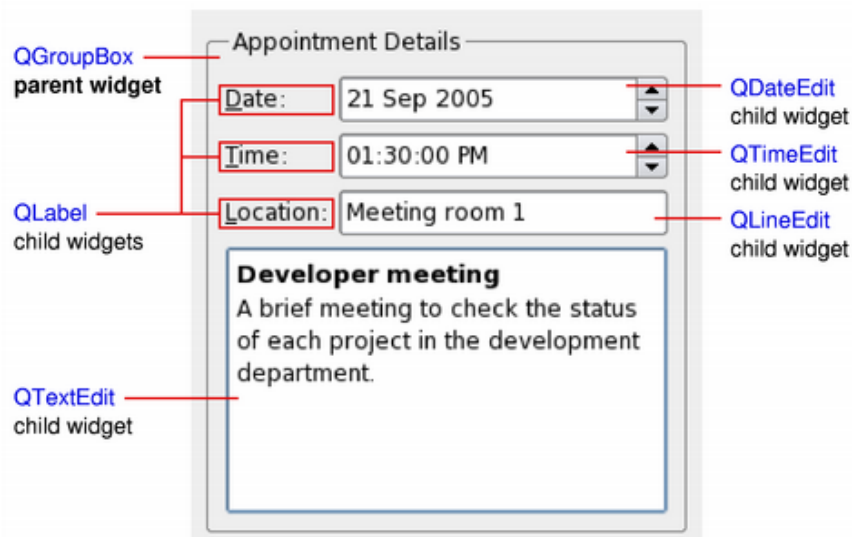
Para este capítulo abordamos los beneficios y las ventajas de este framework para la creación de interfaces de usuario y de forma más específica el escenario para la visualización del módulo de Modo M, explicando la jerarquía que se utiliza en este framework para la presentación visual ante el usuario.

Qt es un Framework multiplataforma orientado a objetos que es ampliamente utilizado para el desarrollo de programas que utilicen una interfaz de usuario. De forma nativa este framework utiliza el lenguaje de programación C++ pero puede ser implementado por otros lenguajes de programación por medio de bindings. Python no es la excepción, Qt dispone de tres bindings para dicho lenguaje, de los cuales PyQt que es un binding GPL/Commercial.

#### Qt Widgets.

El módulo de Qt Widgets provee de un conjunto de elementos de interfaz de usuario para crear interfaces de usuario de estilo de escritorio clásico (ver Figura 10).

Los widgets son los elementos principales para crear interfaces de usuario en Qt, pueden mostrar datos e información de estado, recibir información del usuario y proporcionar un contenedor para otros widgets que deben agruparse. Un widget que no está incrustado en un widget principal se denomina ventana (Qt Group, s. f.).

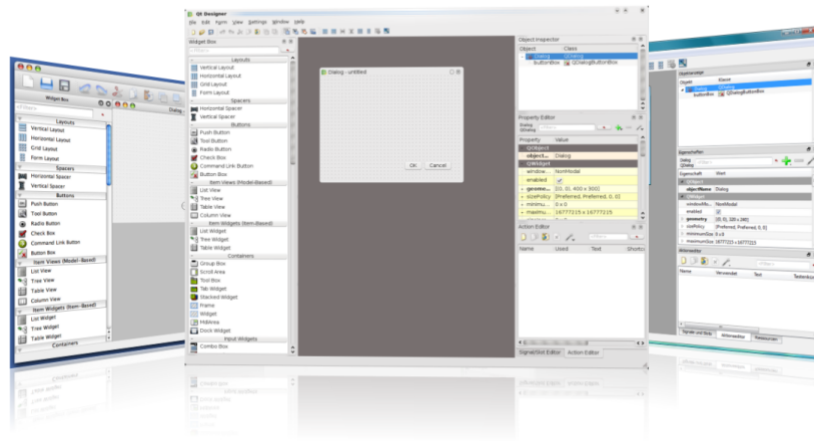


**Figura 10. Elementos principales de un widget en Qt (Tomado de (Qt Group, s. f.)).**

La clase QWidget proporciona la capacidad básica para representar en la pantalla y manejar los eventos de entrada del usuario. Todos los elementos de la interfaz de usuario que proporciona Qt son subclases de QWidget o se usan en conexión con una subclase de QWidget. La creación de widgets personalizados se realiza subclasificando un QWidget o una subclase adecuada y reimplementando los controladores de eventos virtuales (Qt Group, s. f.).

## Qt Designer.

Es una herramienta de Qt para el diseño y construcción de interfaces gráficas de usuario (GUI) con Qt Widgets. Qt Designer implementa una forma para componer y personalizar las ventanas y cuadros de diálogo what-you-see-is-what-you-get (WYSIWYG). Todas las propiedades establecidas en Qt Designer se pueden cambiar dinámicamente dentro del código, además de los complementos personalizados permiten el uso de componentes propios con Qt Designer. Una vista previa del entorno de desarrollo se muestra en la Figura 11 (Qt Group, s. f.).



**Figura 11. Interfaz principal de Qt Designer (Tomado de (Qt Group, s. f.)).**



## **CAPÍTULO 3. Metodología**

---

La importancia del análisis metodológico y concreto del problema condujo a la eficiente construcción de una solución, tomando como marco de referencia el conocimiento empírico de los profesionales del área de la salud y las necesidades presentes que dan pie al desarrollo de este trabajo.

El sistema inteligente para el diagnóstico médico asistido para enfermedades cardiovasculares **SYSCARDIO**, permite analizar la dinámica de las cavidades cardíacas, principalmente aurícula y ventrículo derechos, a partir de imágenes médicas de ecocardiografía, utilizando técnicas de procesamiento de imágenes, visión por computadora y aprendizaje automático.

El sistema dispone de un módulo central en la interfaz de usuario el cual tiene las funcionalidades para:

- **Lectura de Imágenes:** Dada la necesidad de la lectura de las imágenes de ultrasonido, se desarrolló la funcionalidad para el procesamiento de los formatos de imagen en particular utilizados en el ámbito médico cómo lo son el formato BMP, DICOM y MHD. Estos dos últimos también manejan secuencias de ultrasonido.
- **Visualización:** Nos permite poder visualizar las imágenes de los formatos ya especificados, además de poder visualizar secuencias de ultrasonido y videos de ultrasonido para el análisis cualitativo y cuantitativo, está visualización podemos verla en la Figura 12.
- **Interacción del usuario:** La interfaz es bastante intuitiva que permite la fácil interacción del usuario con los componentes de esta, además de la manipulación por parte del usuario de las imágenes que se despliegan en la interfaz.

- **Despliegue de resultados:** Es posible desplegar en la interfaz resultados obtenidos en el análisis cualitativo y cuantitativo de las cavidades cardiacas y sus correspondientes propiedades y características.



Figura 12. Visualización de imagen de ultrasonido en formato de video por medio del módulo Open de SYSCARDIO.

## CAPÍTULO 3.1.

### Creación de módulo M mode

---

Para el cálculo y obtención de la imagen de **M mode** una vez desplegada la imagen de ultrasonido en la interfaz de usuario del sistema **SYSCARDIO**, se requirió emular el haz que atraviesa las distintas estructuras cardiacas mostradas en la imagen, para esta tarea se siguió la siguiente metodología:

1. Crear y habilitar un modo de interacción entre el dispositivo de entrada del usuario (mouse) y la imagen desplegada en el sistema SYSCARDIO, dotando de la capacidad para trazar líneas rectas sobre la superficie de la imagen.

2. Obtener la secuencia de puntos que constituyen la línea trazada, seguido de la depuración de este conjunto de puntos, es decir, la eliminación de puntos repetidos.
3. Obtener las coordenadas del punto inicial y el punto final de la línea.
4. Una vez obtenidas las coordenadas del primer y último punto de la recta, es posible obtener la pendiente de la recta.
5. Obteniendo la pendiente es posible calcular el ángulo de inclinación de la recta.
6. Validando el ángulo de inclinación de la recta, se asignan los valores para el espaciado de la imagen de **M mode**.
7. La construcción de la imagen de **M mode**, se realiza tomando en cuenta elementos como una curva polinomial (paramétrica) por partes. y mediante un método de “Teselado” para el recubrimiento de la imagen por medio de patrones de figuras.

En esta sección se profundiza en la metodología antes mencionada, en el proceso de construcción y desarrollo del módulo de **M mode** con base en una lógica matemática aterrizada a código de programación de alto nivel. La creación de cada clase del código se basa en un paradigma orientado a objetos.

La versión de **VTK** que fue implementada para la realización del módulo de **M mode** es la versión **9.0.1**. debido a su estabilidad y compatibilidad en conjunto con otras librerías implementadas en el desarrollo de las clases que constituyen la estructura del código del módulo.

La versión que fue utilizada para la realización de la interfaz de usuario del sistema **SYSCARDIO** y específicamente el módulo de **M mode** es la versión de Qt 5 con su correspondiente binding **PyQt5**.

Las principales librerías que se implementaron en la realización del módulo de **M mode** son las siguientes:

- VTK
- PyQt5 : QtCore, QtGui, QtWidgets
- Numpy
- Math
- PyQt5.QtWidgets : QFileDialog
- QVTKRenderWindowInteractor

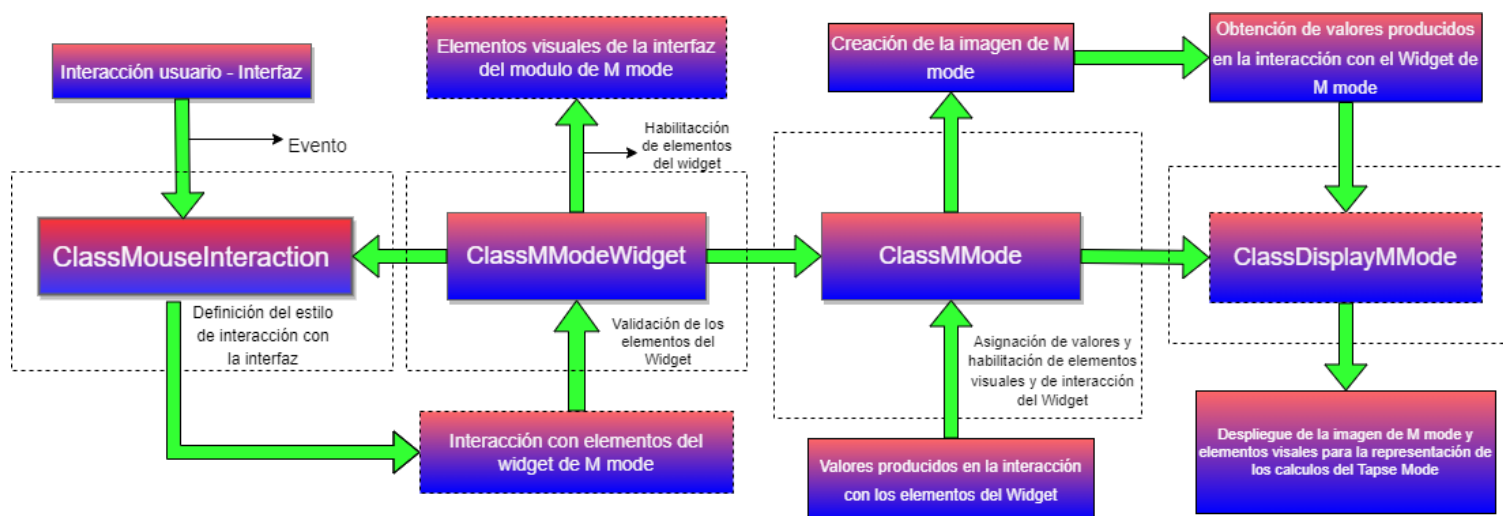
A continuación se detalla cada una de las clases que componen el módulo de M mode, con una descripción general y un diagrama que muestra la estructura de los métodos de la clase con sus respectivos parámetros de entrada. Por medio de estos recursos visuales se tiene cómo objetivo el mostrar la composición de lo que en conjunto conforman el núcleo funcional de este módulo de M mode.

En un inicio para el desarrollo del módulo de M mode fue importante considerar el hecho de que debido a que se utilizó una librería cómo lo es VTK es necesario definir uno o más estilos de interacción con la interfaz de usuario, dependiendo de la tarea que este quiera realizar.

Debido a la forma en cómo funciona una herramienta cómo lo es Qt para el desarrollo de la interfaz de usuario, es necesaria una representación codificada de los componentes de dicha interfaz, además de la funcionalidad empleada por cada uno de estos componentes, esto dió pie a la organización modular del código entre la representación codificada de los componentes visuales y la funcionalidad de los mismos.

Otro punto importante a destacar es la forma en cómo se presentan los resultados del procesamiento de las tareas del módulo de M mode, es decir, la forma en cómo el usuario visualiza los resultados en la interfaz de usuario, esto requirió de la organización del código para el despliegue de los resultados, esto se detalla más adelante en la clase **ClassDisplayMMode**.

A continuación en la Figura 13 se presenta un diagrama general en el cual se representa la relación entre las clases que componen el módulo de M mode.



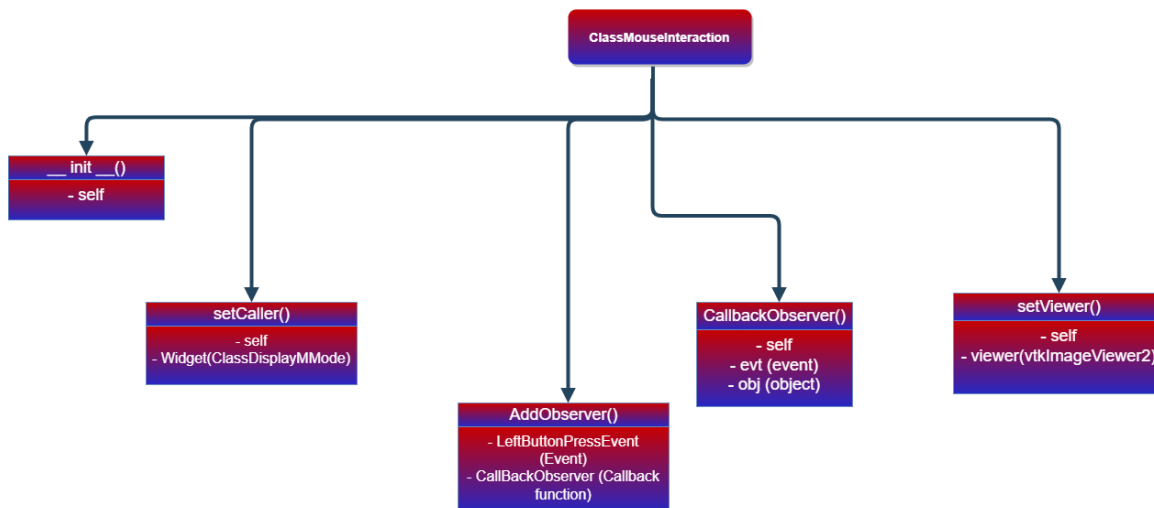
**Figura 13. Diagrama de la estructura general de las clases que componen el módulo de M mode.**

En la creación de las clases y funciones correspondientes con el módulo de **M mode** se presentaron diversos retos en el aspecto técnico. El estilo de interacción en el sistema de software de VTK es la forma en la que se definen las rutinas de control de movimiento y se define una interfaz impulsada por eventos.

La clase **ClassMouseInteraction** hereda de la clase **vtkInteractorStyleTrackballCamera** la cual es una clase definida en el código fuente de la librería VTK. Esta clase se implementó para crear un estilo de interacción con propósitos específicos para hacer posible la creación de la imagen en modo M.

Los métodos definidos en esta clase hacen posible la interacción entre uno de los métodos de entrada el cual es el mouse y la interfaz del usuario, definiendo un “observador” que es un elemento en el código de programación que monitorea los “eventos” que sucedan en la interfaz, estos eventos son las interacciones del usuario con la imagen desplegada en el sistema.

En la Figura 14 se muestra el diagrama de la estructura de la clase **ClassMouseInteraction**, los métodos que la componen y los parámetros que recibe cada uno de estos métodos.



**Figura 14. Estructura de la clase ClassMouseInteraction.**

El propósito principal de la clase **ClassMModeWidget** es la definición, caracterización visual y funcional del Widget de M mode. En la Figura 17 mediante un diagrama se muestra la estructura de esta clase.

Esta clase es producto de la conversión del archivo con extensión .ui que genera el framework de Qt en el diseño de la interfaz de usuario del módulo de M mode. A continuación, se muestra el proceso de conversión de un archivo de interfaz de usuario generado por medio de Qt Designer a un archivo de extensión .py que es un archivo de código Python.

Por medio de la sintaxis del comando de conversión entre estos archivos que es la siguiente:

```
pyuic5 -x file.ui -o file.py
```

Se obtiene un archivo con el código python correspondiente con la estructura de la interfaz de usuario definida en el archivo UI, en la Figura 15 se muestra un ejemplo del código python generado en dicha conversión.

```

class Ui_Mmode(object):
    def setupUi(self, Mmode):
        Mmode.setObjectName("Mmode")
        Mmode.resize(400, 300)
        self.MmodeWidget = QtWidgets.QWidget(Mmode)
        self.MmodeWidget.setGeometry(QtCore.QRect(20, 0, 221, 200))
        self.MmodeWidget.setObjectName("verticalLayoutWidget")

        self.gridLayout = QtWidgets.QGridLayout(self.MmodeWidget)
        #self.verticalLayout = QtWidgets.QVBoxLayout(self.verticalLayoutWidget)
        self.gridLayout.setObjectName("gridLayout")
        #self.verticalLayout.setContentsMargins(11,40,11,40)
        self.verticalLayout = QtWidgets.QVBoxLayout()
        self.verticalLayout.setObjectName("verticalLayout")

        self.AnnotateButton = QtWidgets.QPushButton(Mmode)
        #self.AnnotateButton.setMouseTracking(True)
        self.AnnotateButton.setAutoFillBackground(True)
        self.AnnotateButton.setCheckable(True)
        self.AnnotateButton.setChecked(False)
        self.AnnotateButton.setObjectName("AnnotateButton")
        self.verticalLayout.addWidget(self.AnnotateButton)

        self.ShowMModeChkBox = QtWidgets.QCheckBox(Mmode)
        #self.ShowMModeChkBox.setMouseTracking(True)
        self.ShowMModeChkBox.setAutoFillBackground(True)
        self.ShowMModeChkBox.setObjectName("ShowMModeChkBox")
        self.verticalLayout.addWidget(self.ShowMModeChkBox)

```

Figura 15. Estructura definida en la conversión del archivo a código Python de la clase ClassMModeWidget.

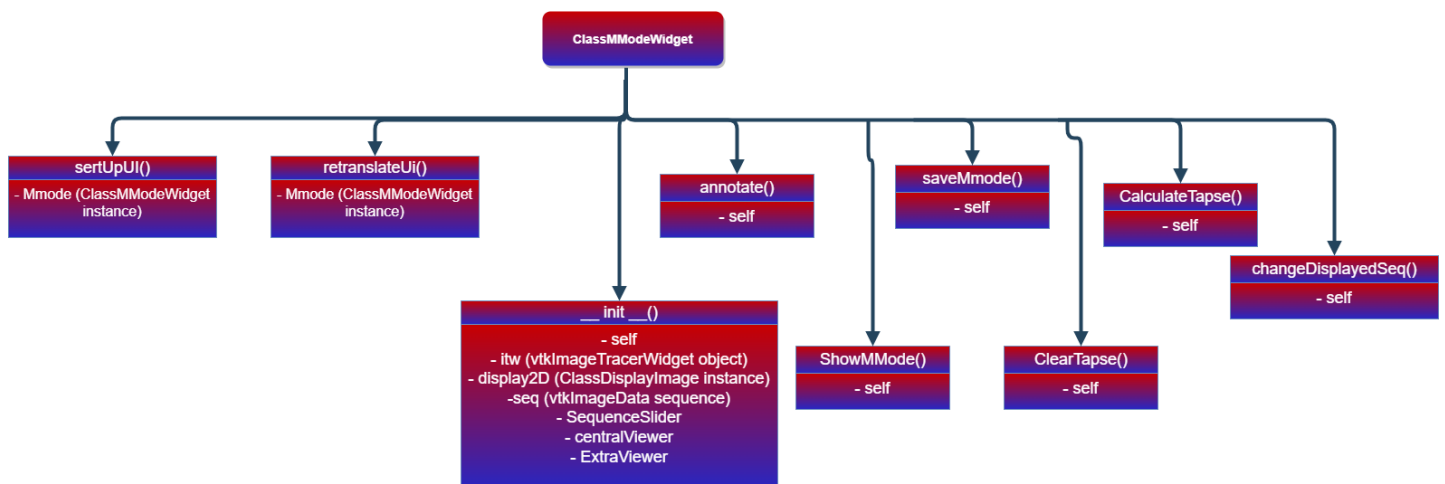
```

self.AnnotateButton.setText(_translate("Mmode", "Annotate"))
self.ShowMModeChkBox.setText(_translate("Mmode", "Show M mode"))
self.TapseModeButton.setText(_translate("Mmode", "Tapse Mode"))

```

Figura 16. Traducción de los elementos de interfaz de usuario a variables en código Python.

En esta clase se encuentran implementados los métodos que obtienen y validan el valor de los elementos gráficos, dependiendo del valor que tengan se habilitan o deshabilitan, por ejemplo el que da la capacidad al usuario poder trazar líneas rectas y líneas poligonales sobre la imagen desplegada, esto con el propósito de marcar áreas de interés del usuario en la imagen de ultrasonido o secuencia de tiempo de ultrasonido. La traducción de dichos elementos gráficos a código python se puede ver ejemplificada en la Figura 16.



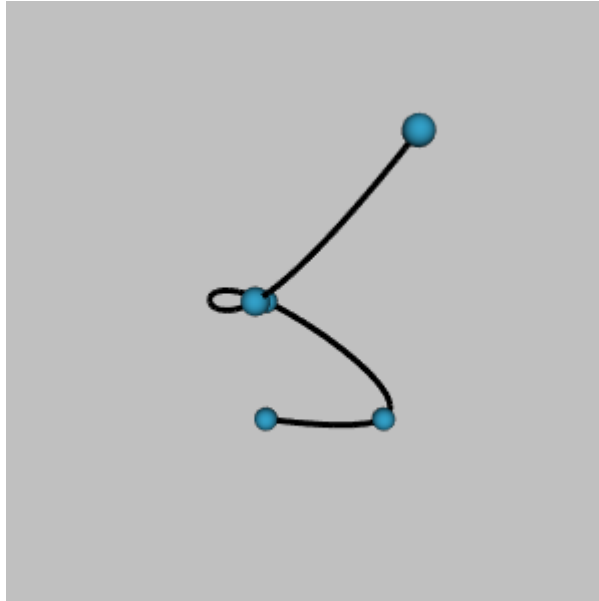
### Figura 17. Estructura de la clase ClassMModeWidget.

El propósito principal de la clase **ClassMMode** es el de la creación de la imagen en modo M, para ello dispone de parámetros y de una serie de métodos que se encuentran organizados en la representación gráfica de la Figura 21.

La construcción de la imagen de **modo M** implica una serie de procesos a partir del trazo que realiza el usuario sobre la imagen de secuencia de ultrasonido desplegada en el widget principal. A continuación se describen dichos procesos:

- Utilizando la clase `vtkParametricSpline` se crea un **Spline paramétrico** que en los subcampos de la informática del diseño asistido por computadora y los gráficos por computadora, el término **spline** se refiere con mayor frecuencia a una **curva polinomial (paramétrica) por partes**, visto en su representación gráfica en la Figura 18.
- La clase `vtkParametricFunctionSource` se utiliza para el “teselado” de funciones paramétricas, el **teselado** hace referencia al recubrimiento completo de una superficie plana por medio de ciertos patrones de figuras o de una cierta regularidad. Para el caso de uso de esta clase se debe especificar la cantidad de puntos que se requieren en las direcciones de coordenadas paramétricas.
- Dicho lo anterior, el spline construido a partir de los puntos obtenidos de la ruta del trazo que el usuario efectúa en la imagen, se asigna a la función paramétrica, para que posteriormente se lleve a cabo el proceso de eliminación de puntos repetidos.





**Figura 18. Representación visual de un spline paramétrico de acuerdo con la documentación de vtk-examples.**

A continuación en las fórmulas (1) y (2) se presenta el modelado de las operaciones a realizar para la eliminación de los puntos repetidos obtenidos de la ruta del trazo del usuario en el eje X y el eje Y:

$$matrixX = roundOut\left(\frac{parametricFuncSource[0]}{spacing[0] + 0.5}\right) \quad (1)$$

$$matrixY = roundOut\left(\frac{parametricFuncSource[1]}{spacing[1] + 0.5}\right) \quad (2)$$

De acuerdo con las fórmulas (1) y (2) las variables descritas en estas hacen referencia a lo siguiente:

- **matrixX:** Esta variable hace referencia a la matriz de puntos correspondientes al eje X que se genera después del cálculo de la fórmula.
- **matrixY:** Esta variable hace referencia a la matriz de puntos correspondientes al eje Y que se genera después del cálculo de la fórmula.
- **roundOut:** Esta operación se encarga del redondeo hacia abajo al entero más cercano.
- **parametricFuncSource:** Esta variable hace referencia al arreglo de puntos en ya sea en el eje X (`parametricFuncSource[0]`) o en el eje Y (`parametricFuncSource[1]`) de la función paramétrica descrita anteriormente.

- spacing: Hablando en términos de VTK el spacing de una imagen definida en este kit de herramientas de visualización, hace referencia al espaciado por “voxel” que es es la unidad cúbica que compone un objeto tridimensional, que componen dicha imagen, para este caso en específico de la fórmula se utiliza el espaciado de la imagen por voxel en el eje X (spacing[0]) y en el eje Y (spacing[1]).

Nota: La adición de un extra en este caso de 0.5 tiene cómo propósito el redondeo hacia arriba de la medida obtenida en el espaciado de la imagen de ultrasonido desplegada en la interfaz de usuario.

Una vez obtenidos los puntos en la matriz de puntos tanto en el eje X cómo en el eje Y, se crea un objeto que es de tipo “vtkPoints” está hace referencia al tipo de puntos que la librería VTK reconoce, y en este objeto se inserta cada punto obtenido por medio de un identificador.

El cálculo del ángulo de inclinación de la línea para el Modo M parte de la obtención del número total de puntos que hay en la línea que trazó el usuario una vez realizada su depuración, para posteriormente obtener el primero y el último de dichos puntos y de esta forma poder aplicar la fórmula que obtiene la pendiente de una recta por medio de las coordenadas X e Y de dos puntos, la cual se muestra en la fórmula (3). El resultado nos permite obtener el ángulo teta de dicha recta.

$$m = \frac{y_2 - y_1}{x_2 - x_1} \quad (3)$$

Una vez que se ha calculado la pendiente de la recta, se necesita validar el ángulo de inclinación para la creación de la imagen de M mode, este ángulo determina el “espaciado (spacing)” correcto para la imagen y se puede obtener por medio de la fórmula matemática cómo se ve en la fórmula (4). Una vez asignado el “espaciado (spacing)” de la imagen, se realiza la asignación de valores a las dimensiones de la imagen modo M.

$$\theta = \tan^{-1}(m) \quad (4)$$

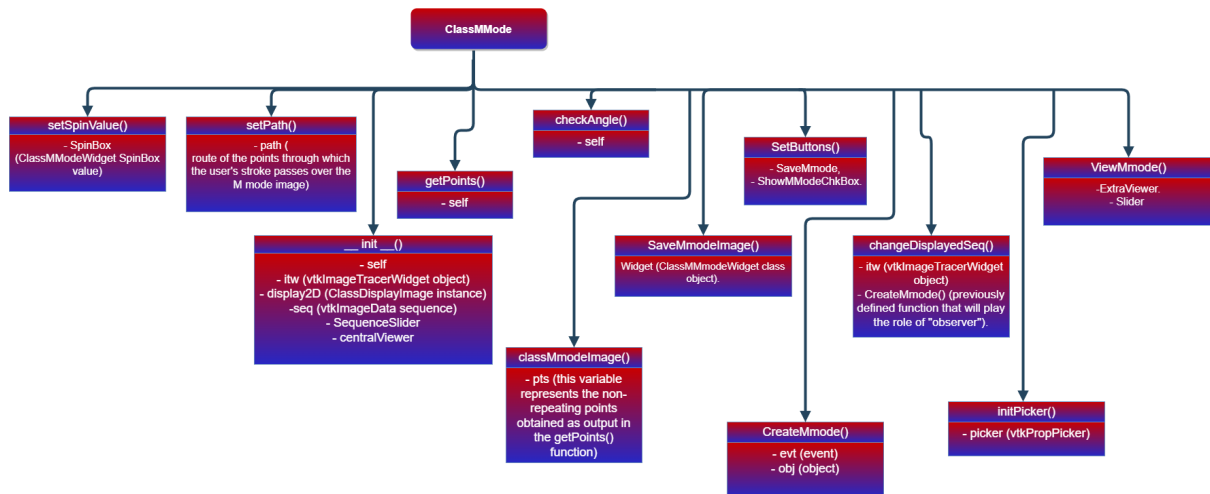


Figura 19. Estructura de la clase ClassMMode.

En la clase **ClassDisplayMMode** se creó para definir las variables, métodos e instancias de las clases auxiliares, para poder lograr la visualización de la imagen de M mode en la interfaz gráfica del usuario, la estructura de esta clase se representa por medio de la Figura 20.

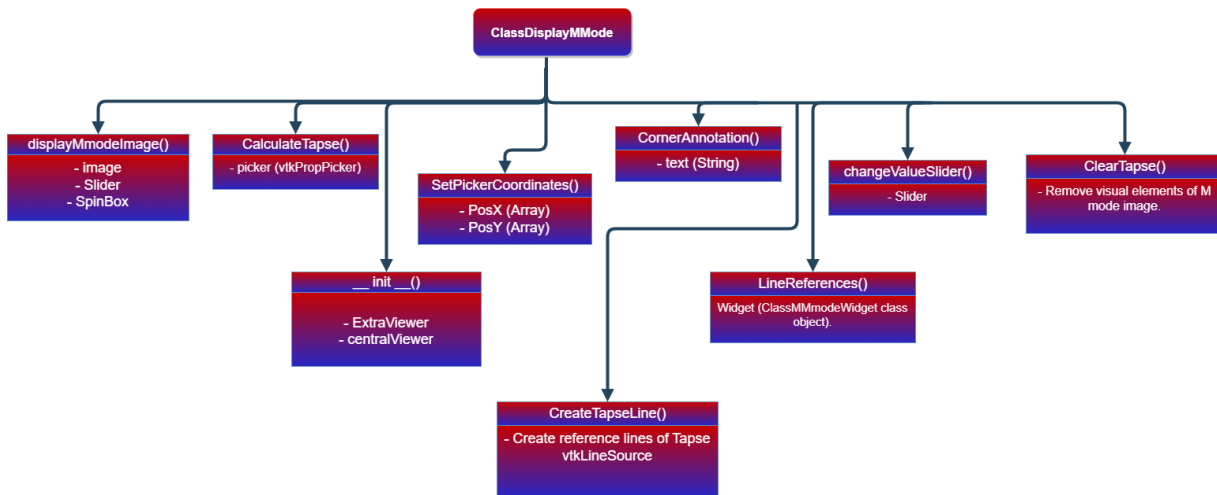
Para el cálculo del **Tapse Mode** se establece el estilo de interacción correspondiente por medio de un objeto de tipo `vtkPropPicker` que se utiliza para seleccionar instancias de tipo `vtkProp3D` de la librería VTK disparando un rayo en una ventana gráfica e interceptando con el cuadro delimitador del actor.

El cálculo de la distancia del **Tapse Mode**, se realiza por medio del cálculo de la distancia entre dos puntos. Este valor se muestra en la interfaz por medio del método **CornerAnnotation()** que tiene como fin mostrar sobre la imagen desplegada anotaciones de los resultados obtenidos. La distancia del Tapse Mode obtenida se muestra en unidades de milímetros.

Cómo referencia visual de los puntos que el usuario marcó en la imagen de M mode, se crean dos líneas de referencia para poder visualizar mejor la distancia entre ambos puntos por medio de la implementación de la clase `vtkLineSource` de la librería VTK, podemos crear una línea definida por dos puntos, estos dos puntos los

obtenemos del arreglo de coordenadas tanto en el eje X cómo en el eje Y obtenidos en el método **SetPickerCoordinates()**.

Se crea una línea horizontal por cada punto en el que el usuario dio un click en la imagen de M mode, esto cómo una referencia para el valor calculado en el Tapse Mode, asignando las propiedades visuales para estas dos líneas.



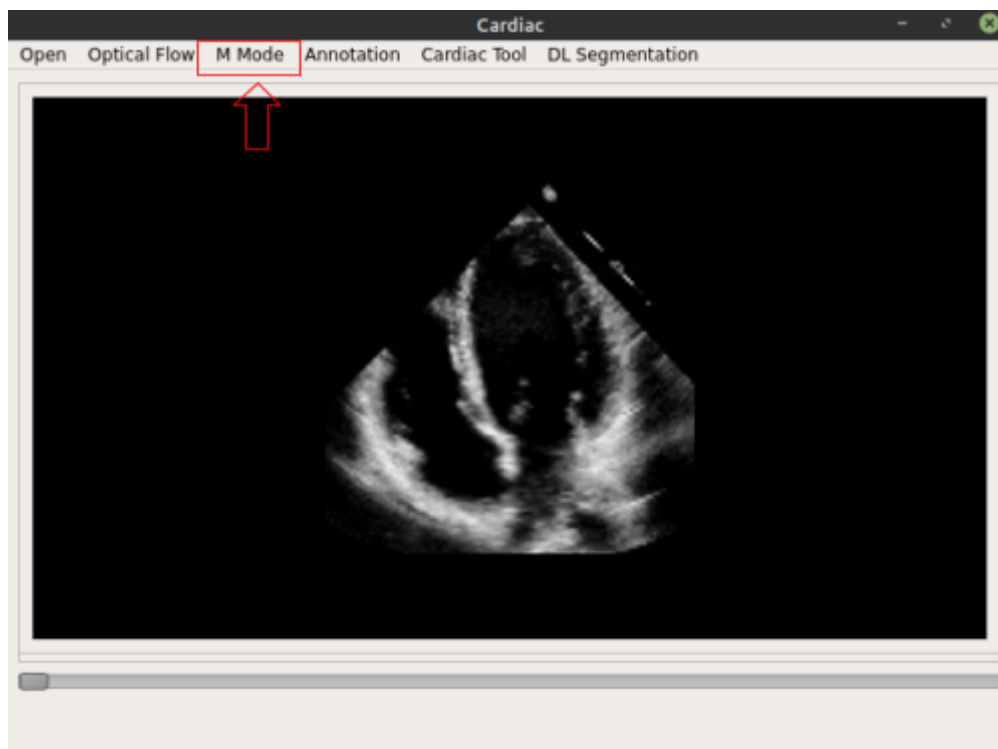
**Figura 20. Estructura de la clase ClassDisplayMMode.**

## CAPÍTULO 4. Pruebas y Resultados

En esta sección se muestra el funcionamiento del módulo de M mode, por medio de la ejecución del sistema SYSCARDIO, y haciendo uso de las herramientas que nos brinda nuestro módulo de M mode, se pueden visualizar los resultados obtenidos en su desarrollo e implementación.

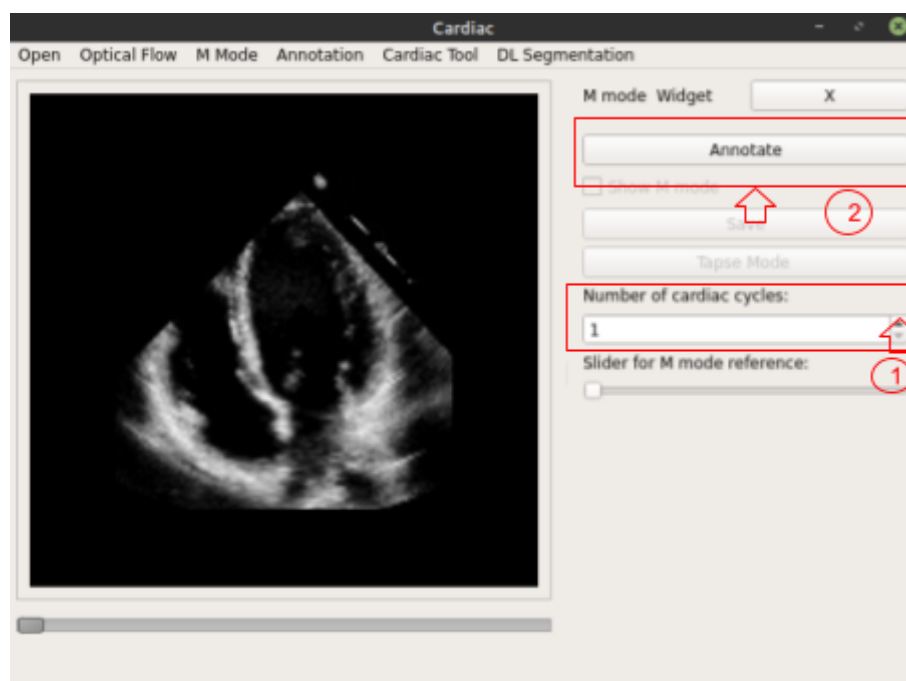
Para la demostración del funcionamiento del módulo de M mode, es necesaria la apertura de una imagen de secuencia o bien un video, ya que este tipo de formatos de imagen se componen por más de un frame, debido a que en su totalidad conforman una secuencia de tiempo, por ello son los formatos ideales para poder obtener la imagen en M mode dada la naturaleza de este modo de

visualización de ultrasonido, esta visualización de la imagen de secuencia se puede apreciar en la Figura 21.



**Figura 21. Secuencia de tipo video desplegada en el Main Window. Barra de menú dónde se puede apreciar la opción para el módulo de M mode.**

Una vez seleccionada la opción del módulo de M mode del menú podemos visualizar la siguiente ventana en la cual podremos seleccionar las herramientas para crear la imagen de M mode, tal cómo se muestra en la Figura 22.



**Figura 22. 1- Se selecciona el número de ciclos cardíacos que se desea visualizar en la imagen de M mode. 2- Se selecciona el botón de Annotate para activar el widget de anotación y poder trazar líneas sobre el video desplegado.**

Para la creación de la imagen de M mode es necesario ubicar en la secuencia de tiempo el movimiento de apertura de la válvula tricúspide, por la cual emulando la técnica de ultrasonido de M mode, se traza una línea recta desde la punta superior de la imagen del ultrasonido, pasando a través del anillo tricúspide y finalizando en la esquina inferior izquierda de la imagen de ultrasonido. Este proceso se puede apreciar en la Figura 23.

**Nota Importante:** Dado que se utiliza la clase `vtkImageTracerWidget` cumple con ciertas especificaciones para las acciones de los botones y las teclas modificadoras, para que sea posible que el usuario logre una línea recta debe de contar con un mouse con “scroll” ya que presionando este en el punto donde se desea iniciar la línea, para terminar la línea se debe presionar este mismo botón en conjunto con la tecla “shift”.



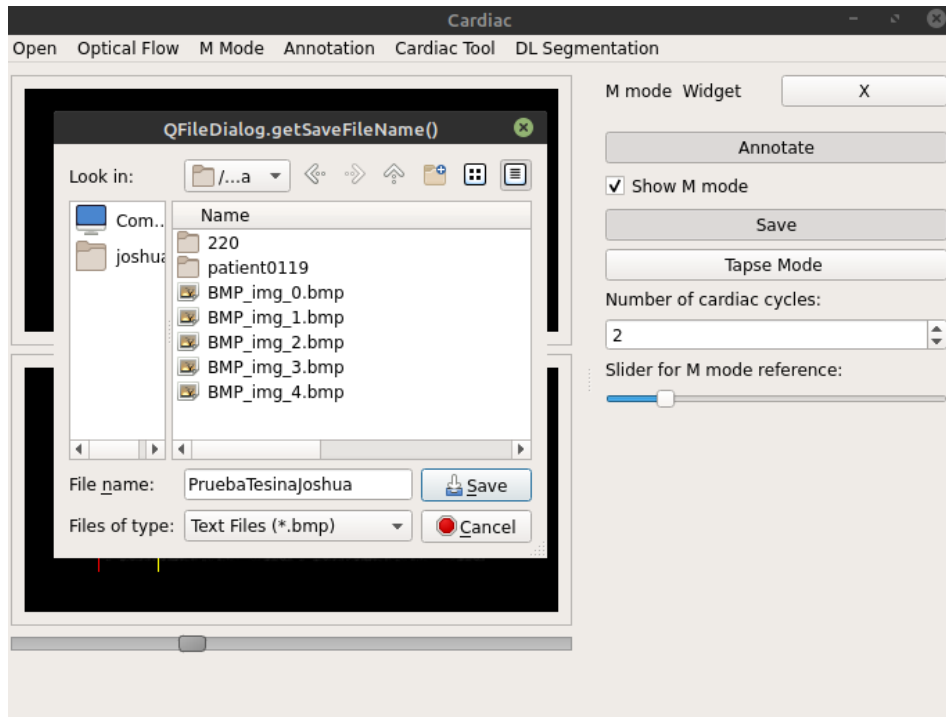
**Figura 23. Trazo de la línea para la creación de la imagen de M mode seleccionando 3 ciclos cardíacos para visualizar en la imagen.**

Una vez realizado el trazo de la línea recta es necesario dar click izquierdo del mouse en el recuadro que se ubica a un lado del texto “Show M mode” para poder visualizar la imagen de M mode que se creó a partir del trazo de la línea recta, tomando cada uno de los pixeles por los que pasa la ruta de la línea recta.

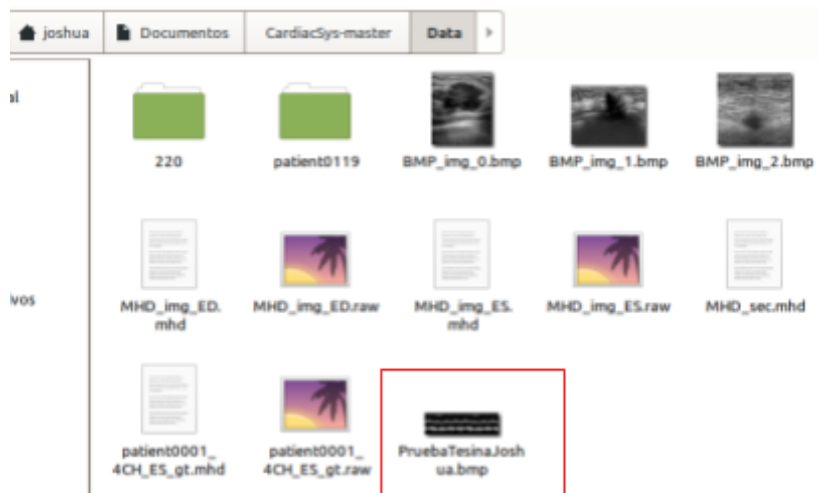


**Figura 24. Imagen de M mode desplegada en su correspondiente frame, desplazando el Slider del módulo de M mode se puede visualizar la segunda línea de referencia en la imagen.**

Cómo es posible apreciar en la Figura 24, además de mostrar la imagen de M mode y sus correspondientes líneas de referencia, una de las cuales es posible desplazar a lo largo de la imagen, por medio del elemento deslizable ubicado abajo del texto "Slider for M mode reference" del módulo de M mode. Se habilita el botón para guardar (Save) la imagen de M mode que al presionarlo despliega una ventana en el widget principal para escribir un nombre para guardar la imagen, este proceso se ve representado en las figuras 25 y 26. Además de habilitar el botón para poder crear el Tapse Mode.

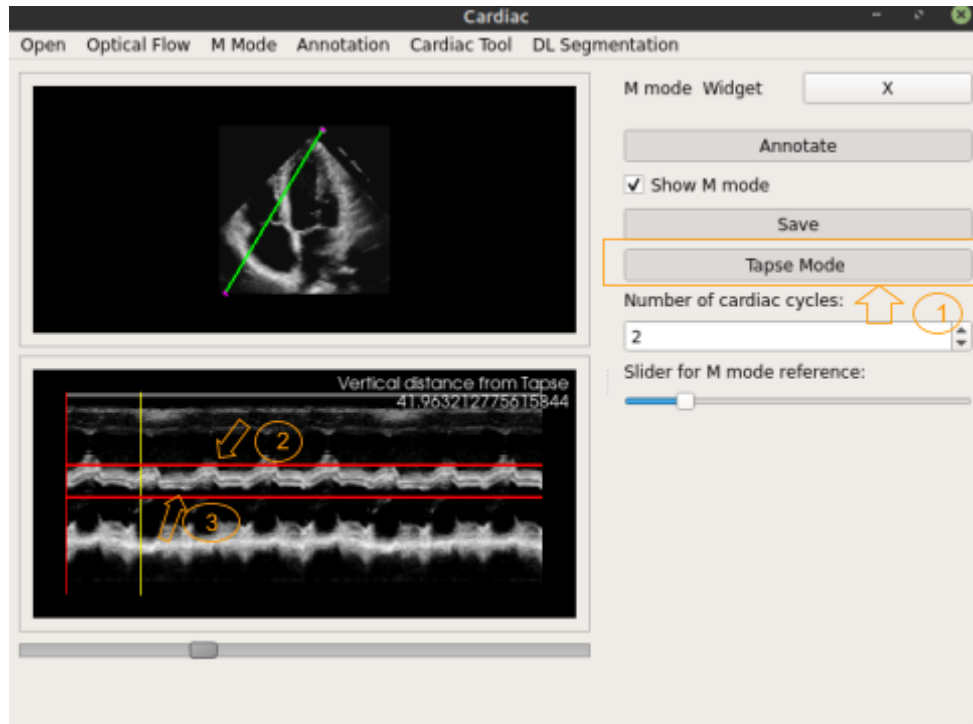


**Figura 25. Ventana emergente para guardar la imagen de M mode en la ubicación local que el usuario destine, con un formato de archivo de tipo bmp. Damos un nombre personalizado al archivo.**



**Figura 26. Generación del archivo bmp en la ruta y con el nombre designado por el usuario.**





**Figura 27. 1- Seleccionamos el botón de Tapse Mode, esto habilitará el estilo de interacción que le permite al usuario indicar con un click la posición inicial y la posición final de la distancia deseada. 2- Seleccionamos el punto inicial del Tapse. 3- Seleccionamos el punto final de la distancia del Tapse en la imagen de M mode.**

Después de que el usuario selecciona los puntos de inicio y fin que marcan la distancia que desea calcular del Tapse mode, dos líneas de referencia se crean a lo ancho de la imagen de M mode, tomando como base las coordenadas de los puntos, de esta forma es posible visualizar mejor la distancia entre ambos puntos tal como se muestra en la Figura 27.

En la parte superior derecha de la imagen de M mode se puede apreciar una etiqueta dónde se hace referencia al valor que está por indicarse abajo de dicha etiqueta, en este caso la distancia vertical del Tapse.

Una vez obtenido el cálculo del Tapse, se puede cerrar la ventana del Widget, presionando el botón con el icono de la “X”, esto cerrará la ventana, y hará un reset de la información de la ejecución del módulo, es decir, si el usuario abre nuevamente el Widget las anotaciones y la imagen de M mode ya no aparecerán, al igual que el cálculo del Tapse, por ello es importante que, una vez ejecutado el módulo, se guarden los datos pertinentes y de utilidad para el médico especialista.

También se creó como parte de este trabajo el manual de usuario del sistema SYSCARDIO en el capítulo de Anexos de esta tesina.

## CAPÍTULO 5.

### Conclusiones

---

El diseño, desarrollo y prueba del módulo de **M mode** como parte de la funcionalidad del sistema **SYSCARDIO** son reflejo del logro obtenido de acuerdo con el Objetivo establecido en este trabajo escrito. Implicando en sí mismo la muestra de la aplicación de los conocimientos adquiridos a lo largo de la carrera de Ingeniería en Computación.

El desarrollo de este módulo implicó grandes retos académicos y personales poniendo a prueba los conocimientos adquiridos a lo largo de la carrera desde áreas como la programación estructurada y orientada a objetos hasta conocimientos adquiridos en ciencias básicas reflejados en el cálculo del ángulo de inclinación de una línea recta por mencionar solo algunos.

La creación y habilitación de un modo de interacción para el trazado de líneas rectas sobre la imagen de ultrasonido del módulo de **M mode** queda sujeto a ciertas consideraciones y limitaciones de acuerdo con el hardware utilizado para realizar el trazado. Si bien cumple con su objetivo queda abierto a la mejora y optimización de esta tarea eliminando la dependencia actual que existe con el hardware que se utilice en el equipo donde se ejecute el sistema.

En el cálculo del Tapse mode, de acuerdo con los resultados obtenidos se concluye que pueden existir ciertas variaciones con respecto al estándar que se tiene para dicha medida, esto tomando en cuenta el margen de error debido a la naturaleza del tipo de sistema del cual es parte el módulo **M mode**.

La funcionalidad del módulo **M mode** cumple con los objetivos establecidos en el desarrollo del sistema **SYSCARDIO**, brindando los elementos necesarios al médico especialista que en conjunto con su expertise y conocimiento en el área contribuyen a la generación de un diagnóstico oportuno y confiable.

## Referencias

---

1. Fuentes Molina Edgar Dr, Lowenstein Jorge Dr. (2010). Guías para la Evaluación Ecocardiográfica de las Cámaras Derechas en el Adulto: Un Reporte de la Asociación Americana de Ecocardiografía. *Revista de la Sociedad Americana de Ecocardiografía*, pp. 23 - 40. Recuperado 19 de Febrero de 2022.
2. Madrid Dour, Eddy Alicia. (2015). Hipertensión Pulmonar. *Ecocardiografía Centro Médico Nacional 20 de Noviembre*. Recuperado el 7 de abril de 2022.
3. Lang RM, Bierig M, Devereux RB, Flachskampf FA, Foster E, Pellikka PA. (2005). Recommendations for chamber quantification: a report from the American Society of Echocardiography's Guidelines and Standards Committee and the Chamber Quantification Writing Group, developed in conjunction with the European Association of Echocardiography. *European Society of Cardiology. J Am Soc Echocardiogr* 2005;18:1440-63. Recuperado el 7 de abril de 2022.
4. D. Thomas James, MD, B. Popovic Zoran. (2006, Noviembre). Assessment of Left Ventricular Function by Cardiac Ultrasound. *Journal of the American College of Cardiology*, 48(10), 2012–2025. Recuperado el 10 de abril de 2022.
5. Ordoñez Vázquez, O. V. L., Montes Castillo, M. C. C., Mora Soto M. S. J. R., Sánchez Hernández S. H. J. D. (2015). Hipertensión arterial pulmonar. *El Residente*, 10(1), medigraphic. Recuperado el 13 de abril del 2022.
6. Marcus D. H., Kenneth M. M., Aashish C., Lisa S. A. (2015). The Visualization Toolkit (VTK): Rewriting the rendering code for modern graphics cards. *SoftwareX*. Volumes 1–2. 9-12. Recuperado el 13 de abril del 2022.
7. Sandoval Zarate, Julio. (2006). Hipertensión arterial pulmonar. *Archivos de cardiología de México*, 76(Supl. 2), 69-75. Recuperado el 13 de abril de 2022.
8. Bouzas B. (2009). *Ventrículo derecho pulmonar y sistémico*. *Revista Española de Cardiología Suplementos*. 9(5). 23-29. Recuperado el 13 de abril del 2023.
9. López Nilson MD, Escobar Carlos I. MD, López Francisco MD, Durango Luisa MD, Fernández Ricardo MD, Franco Gloria MD. (2012). *Función longitudinal del ventrículo derecho estimada por el desplazamiento sistólico del anillo tricúspide (TAPSE) en 557 pacientes sanos*. *Revista Colombiana de Cardiología*. 19(6). 281-286. Recuperado el 16 de abril del 2022.

10. Python T.M. (2022). *What is Python?*. Python Official Documentation. Python Official Site. Recuperado 17 de abril de 2022 de <https://www.python.org/doc/essays/blurbl/>
11. Gasser, G. R. (2010). *VTK Basics*. TechWeb. Boston University Information Services & Technology. Recuperado 17 de abril de 2022, de <https://www.bu.edu/tech/support/research/software-and-programming/software-and-applications/rcs-software-packages/vtk/>
12. IMSS. (2017). *La Hipertensión Arterial de la población en México, una de las más altas del Mundo*. Gobierno de México: IMSS. Recuperado 17 de abril de 2022, de <http://www.imss.gob.mx/prensa/archivo/201707/203#:~:text=M%C3%A9xico%20registra%20la%20prevalencia%20m%C3%A1s,de%20que%20se%20eleva%20la>
13. Binder, B. T. (2020). *M-Mode ultrasound imaging*. 123 sonography, 10(1), 123 sonography. Recuperado 18 de abril de 2022.
14. Dhawan, I., Makhija, N., Choudhury, M., & Choudhury, A. (2019). *Modified Tricuspid Annular Plane Systolic Excursion for Assessment of Right Ventricular Systolic Function*. Journal of cardiovascular imaging, 27(1), 24–33. Recuperado 19 de abril de 2022
15. García Fernández, G. F. M. A., & Zamorano, Z. J. L. (2020). *Ecocardiografía con Técnica Modo M*. Sociedad Española de Imagen Cardíaca. Recuperado 19 de abril de 2022.
16. Omerovic Silvan. (2022, Mayo 8). *Echocardiogram*. STATPEARLS: Medical Reference. Recuperado el 22 de abril de 2022.
17. Singh, S., & Goyal, A. (2007). The origin of echocardiography: a tribute to Inge Edler. *Texas Heart Institute journal*, 34(4), 431–438. Recuperado el 22 de abril de 2022.
18. Libby, L. P. (2021) *Braunwald's Heart Disease: A Textbook of Cardiovascular Medicine*. Clinicalkey, 12, Elsevier. Recuperado el 22 de abril de 2022.
19. Feigenbaum, F. H. (1996). Evolution of Echocardiography. *Circulation*, 93(7). 1321-1327. Recuperado el 25 de abril de 2022.

20. Ortego, O. A. (2020). ¿Qué es un Ecocardiograma Doppler? PENTA. Recuperado el 28 de abril de 2022.
21. Mohamed, A. A., Arifi, A. A., & Omran, A. (2010). *The basics of echocardiography*. Journal of the Saudi Heart Association, 22(2), 71–76. Recuperado el 28 de abril de 2022.
22. Jafari, Mohammad & Girgis, Hany Younan Azer & Abdi, Amir Hossein & Liao, Zhibin & Pesteie, Mehran & Rohling, Robert & Gin, Kenneth & Tsang, Teresa & Abolmaesumi, Purang. (2019). *Semi-Supervised Learning For Cardiac Left Ventricle Segmentation Using Conditional Deep Generative Models as Prior*. 649-652. Recuperado el 04 de mayo de 2022.
23. Schroeder Will, Martin Ken & Lorensen Bill . (2006). *The Visualization Toolkit (4th ed.)*. Kitware. Recuperado el 05 de mayo de 2022.
24. VTK Wiki. (s.f.). *vtkImageTracerWidget Class Reference*. Public Wiki VTK <https://vtk.org/doc/nightly/html/classvtkImageTracerWidget.html#details>. Recuperado el 17 de mayo de 2022.
25. Qt Group. (s. f.). Qt Documentation. Qt.io Qt Documentation <https://doc.qt.io/> . Recuperado el 17 de mayo de 2022.
26. R. Forfia, F. R. P., R. Fisher, F. R. M., & C. Mathai, C. M. S. (2006). Tricuspid Annular Displacement Predicts Survival in Pulmonary Hypertension. *ATS Journals*. Recuperado el 22 de mayo del 2022.
27. Khandheria Bijoy K. M.D., Seward James B. M.D. & Tajik A. Jamil M.D. (1994, Diciembre). *Transesophageal Echocardiography*. ELSEVIER: Science Direct. *Mayo Clinic Proceedings*. 69(9). 856-863. Recuperado el 24 de Mayo del 2022
28. Ayuela Azcaratea, J. M., Clau Terré, F., Ochagaviac, A., & Vicho Pereira, R. (2012). *Papel de la ecocardiografía en la monitorización hemodinámica de los pacientes críticos*. *Medicina Intensiva*, 36(3), 220-232. Recuperado el 25 de Mayo del 2022.
29. JLS Interactive, LLC. (2010). *M-Mode Hemodynamics*. e-Echocardiography. Recuperado el 27 de Mayo del 2022.

### Sistema de Monitoreo Cardiaco.

#### 1. Objetivo

Se tiene como objetivo el desarrollo e implementación de un sistema de monitoreo cardiaco, mediante un software que permita la visualización, el análisis y la segmentación de imágenes de ultrasonido modo B estáticas y en movimiento. Esto con el fin de facilitar el monitoreo de la función cardiaca y establecer valores biométricos que sean capaces de predecir y diagnosticar pacientes con patologías cardiacas tempranamente.

#### 2. Instalación e implementación del sistema

##### Instalación

- Copie en su directorio correspondiente los archivos del proyecto. Ejemplo en Desktop quedó como:

```
C:\Users\ji\Desktop\Syscardio 1.1\Syscardio 1.1\Syscardio1.1
```

irse a ese directorio

```
cd "C:\Users\ji\Desktop\Syscardio 1.1\Syscardio 1.1\Syscardio1.1"
```

- Teniendo Anaconda instalado, previamente abra el Anaconda Prompt

```
>conda create -n CardiacSys python=3.7
```

- Para activar el ambiente de trabajo

```
>conda activate CardiacSys
```

para desactivarlo

```
>conda deactivate
```

- Instalar las bibliotecas del sistema

```
>pip install -r requirements.txt
```

para correr el programa están estas 2 opciones

>python Main2.py (en el casos de ensamblés)

>python Main.py (en el caso de la unet sola)

## Implementación

Para la implementación del sistema se utilizó como arquitectura de software Modelo-Vista-Controlador, el cual separa los datos, la interfaz de usuario y la lógica del funcionamiento en tres componentes distintos. En este modelo el controlador interactúa como una sección intermedia de código que funge como intermediario de la comunicación entre el modelo (gestiona todos los accesos a la información con la que el modelo opera) y la vista (interfaz de usuario). Este modelo permite separar la lógica del software de la interfaz de usuario para que los desarrolladores puedan modificar cada parte más fácilmente sin afectar a la otra.

Para el desarrollo de software se utilizó el lenguaje de programación Python 3.7, el cual es un lenguaje de propósito general y líder en la implementación de sistemas que hacen uso de *machine learning* y el análisis de datos. Para llevar a cabo la funcionalidad del sistema se hizo uso de las siguientes librerías de acceso libre implementadas en este lenguaje:

Librería	Versión	Función
vtk	8.2.0	Visualización y procesamiento de imágenes en 2D y 3D
pyqt	5.12.3	Desarrollo de interfaces gráficas de usuario
pydicom	1.3.0	Abrir y Escribir archivos DICOM
Pillow	7.0.0	Añade funciones de procesamiento de imágenes a python
openCV	4.1.2.30	Librería para <i>computer vision</i> y <i>machine learning</i>
scikit-image	0.17.2	Colección de algoritmos para procesamiento de imágenes
scikit-learn	0.23.1	Colección de algoritmos de <i>machine learning</i>
joblib	0.13.2	Para la creación de pipelines de trabajo (optimización)

tensorflow	2.1 o 1.15	Librería para la implementación de métodos de <i>deep learning</i>
imgaug	0.3.0	Implementación de métodos de aumento de datos

---

### 3. Descripción del sistema

El sistema de monitoreo cardíaco está integrado a partir de módulos (widgets) independientes los cuales se pueden comunicar entre sí para realizar diferentes tareas. Este sistema cuenta con 5 módulos principales:

- Módulo central
- Anotación manual
- Flujo óptico
- Modo M
- Módulo de cuantificaciones de las cavidades cardíacas (Cardiac Tool)

#### 3.1 Módulo central

El módulo central es la ventana principal de la interfaz de usuario, este se encarga de la lectura, visualización e interacción del usuario con las imágenes, así como el despliegue de resultados y errores. La interfaz gráfica de este módulo se muestra en la Figura A1.





**Figura A1. Partes que conforman al módulo central: 1) Abrir imágenes, 2) módulos de monitoreo cardiaco, 3) visualización de imágenes y 4) slider para cambiar la imagen que se visualiza.**

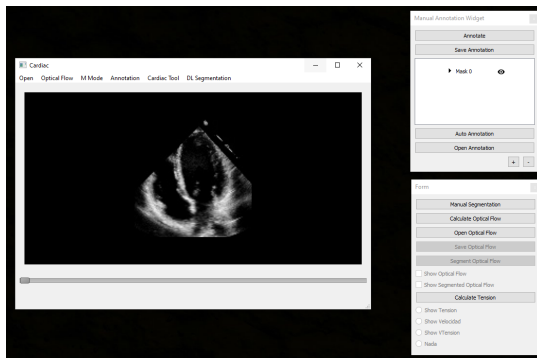
### Lectura de imágenes

El sistema es capaz de leer imágenes de ultrasonido cardiaco de modo B estáticas y en movimiento en diferentes formatos, así como leer múltiples imágenes estáticas al mismo tiempo.

- Formatos 2D: JPG, BMP, PNG, TIFF, MHD, MHA, DICOM.
- Formatos 2D + tiempo: MHD, DICOM, AVI.

### Módulos de monitoreo cardiaco

Botones para abrir los diferentes módulos de monitoreo. Estos módulos están programados como interfaces independientes acoplables al módulo central, como se muestra en la Figura A2. El controlador del módulo central está encargado de la apertura y la interacción entre los módulos.



a)



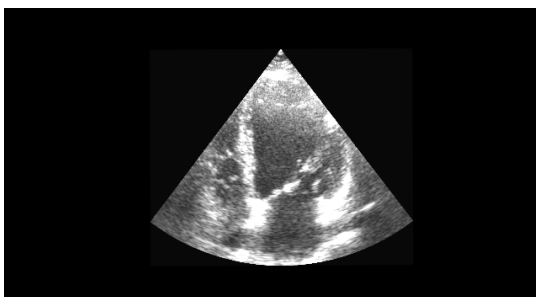
b)

**Figura A2. Módulos de monitoreo cardíaco: a) módulos independientes y b) módulos acoplados en el módulo central**

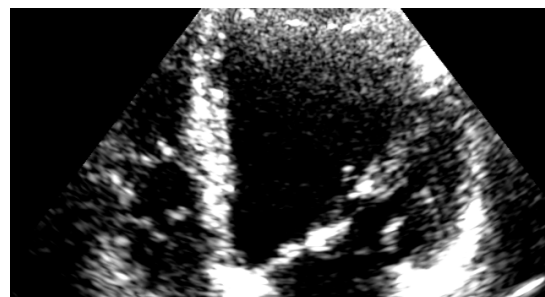
### Visualización de imágenes

La visualización de imágenes se realiza mediante un objeto QVTK, el cual es un módulo de integración de VTK sobre Qt, permitiendo la manipulación de la cámara de vista y dando oportunidad a los médicos para mejorar la visualización de las estructuras deseadas. Las siguiente son algunas de las acciones que este módulo permite sobre la visualización:

- Botón izquierdo: cambio de intensidades de gris en la visualización.
- Botón medio: planea la cámara.
- Botón derecho: aleja y acerca la cámara a la imagen.



a)



b)

**Figura A3. Interacción de usuario con la imagen: a) imagen abierta y b) imagen con zoom y mejora de contraste.**

### 3.2 Clase Main

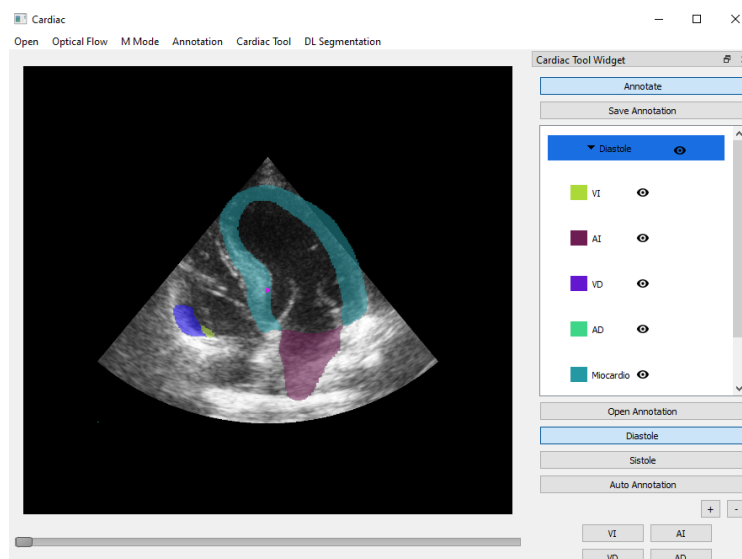
Con motivo de mejorar la funcionalidad y la versatilidad de la herramienta se incorporó en la programación una clase Main. Esta permite reducir los tiempos de espera mientras se cargan las diferentes funcionalidades de la herramienta, también permite abrir nuevas imágenes o cambiar de imagen al mismo tiempo que permite cambiar de funcionalidad sin tener que cerrar y abrir la herramienta.

Para la reducción en el tiempo de carga, la clase contiene los modelos de redes neuronales que son cargados cada vez que la herramienta es iniciada. Si bien este proceso demora unos segundos, solo será necesario esperar este tiempo al iniciar la herramienta en lugar de tener que esperar cada vez que sea utilizada alguna de las funcionalidades que hacen uso de las redes neuronales para la segmentación automática. La herramienta hace uso de nueve redes neuronales, estas se forman de la combinación de tres modelos base y de tres backbones. En la Tabla A1 se muestran los modelos base y los backbones que fueron utilizados.

**Tabla A1. Modelos base y backbones de las redes neuronales**

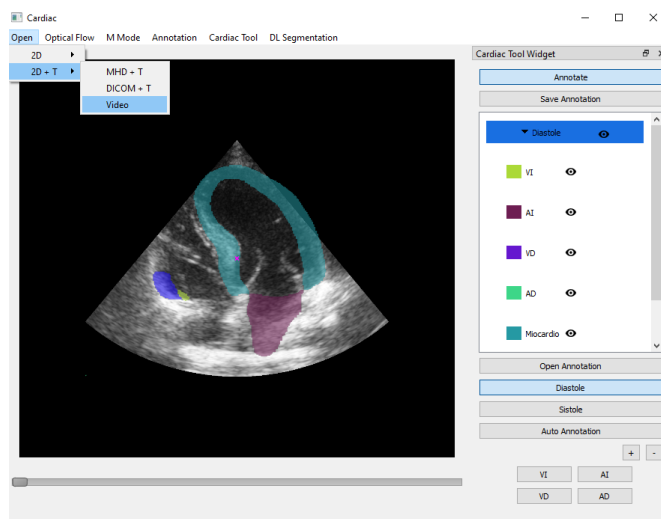
Modelos base	Backbones
Unet	vgg16
Linknet	resnet34
FPN	resnet101

Para poder cargar una nueva imagen, partimos de tener una imagen previamente cargada y tener algunas funciones de la herramienta en uso.



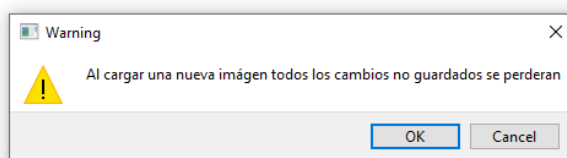
**Figura A4. Muestra de funcionamiento de la herramienta con una imagen previamente cargada y anotaciones realizadas.**

De ahí, solo es necesario seleccionar el botón Open y seleccionar el nuevo tipo de imagen que desee abrir.



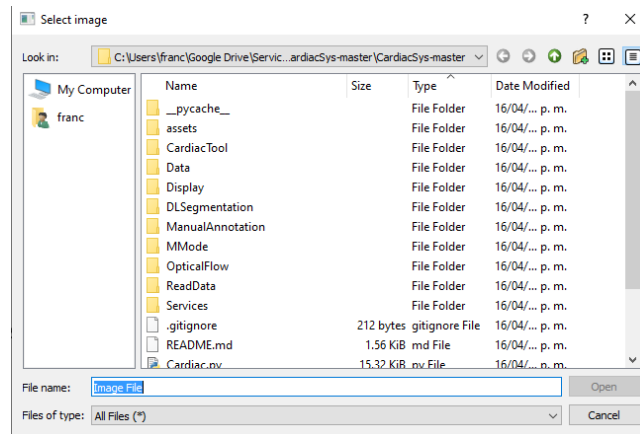
**Figura A5. Selección de botón Open y del tipo de imagen para cargar una nueva imagen.**

Posteriormente, la aplicación arroja un mensaje como el siguiente.



**Figura A6. Ventana de advertencia sobre cambios no guardados.**

Si el usuario desea preservar su trabajo deberá guardarlo, el procedimiento para guardar anotaciones se muestra en un punto posterior en este documento. Si el usuario ha guardado todo, o no desea preservar su trabajo puede dar click en el botón de "OK" y continuar. Acto siguiente se abrirá una ventana para seleccionar el nuevo archivo para cargar la nueva imagen.



**Figura A7. Ventana para seleccionar la nueva imagen a cargar.**

Tras haber seleccionado el archivo, este se mostrará como en la siguiente imagen. Es preciso notar que la herramienta partirá de su configuración inicial por lo que ningún widget de la herramienta estará abierto y todo el trabajo previo habrá desaparecido.



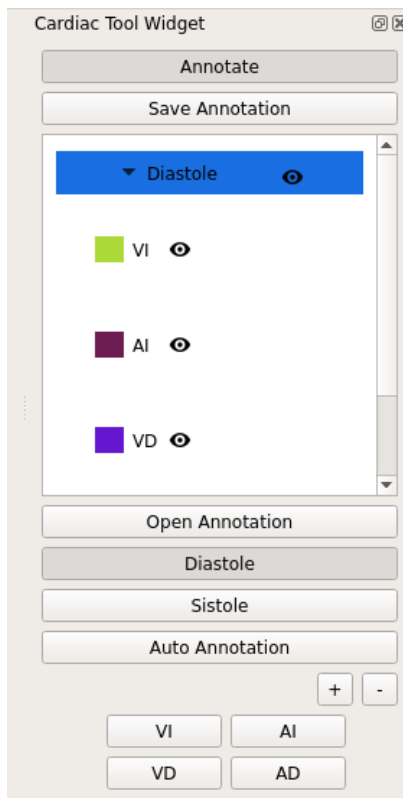
**Figura A8. Nueva imagen cargada y herramienta con configuración base**

Estas dos características permiten un mejor y más veloz funcionamiento de la herramienta.

### 3.3 Módulo de cuantificación de las cavidades cardíacas (Cardiac Tool)

Este módulo permite al médico realizar anotaciones manuales de diferentes estructuras sobre imágenes modo B. Este módulo puede trabajar con imágenes estáticas, múltiples imágenes estáticas e imágenes en movimiento (secuencias de tiempo). La anotación de estructuras cardíacas manualmente tiene dos principales objetos: 1) es permitir al médico realizar cálculos cuantitativos en diferentes momentos del ciclo cardíaco en diferentes estructuras para así obtener mediciones que permitan la evaluación de las estructuras, como lo es el caso del cálculo de área fraccional en la aurícula derecha o la fracción de eyección en el ventrículo izquierdo; y 2) la anotación de imágenes permite la adquisición de datos de entrenamiento necesarios para el entrenamiento de modelos de *machine learning* que sean capaces de realizar la segmentación automática de diferentes estructuras cardíacas.

En la Figura A9 se muestra la interfaz gráfica del módulo de cuantificación de las cavidades cardíacas, así como las funciones que realiza.



**Figura A9. Cardiac Tool widget**

- **Annotate:**  
Habilita la herramienta de segmentación manual.

- **Save Annotation:** Salva la anotación manual.

- **Masks:**  
Máscaras de anotación realizadas sobre la imagen, se pueden

realizar varias máscaras de segmentación

sobre cada imagen. Este a su vez permite la eliminación de anotaciones incorrectas.

- **Auto Annotation:** realiza la segmentación automática de las cavidades para el frame mostrado.
- **Open Annotation:** permite abrir o importar máscaras junto con sus anotaciones que fueron previamente guardadas.
- **Diástole, Sístole:** crea una máscara específica para el período de diástole/sístole, crea cuatro anotaciones de ambos ventrículos y aurículas respectivamente.
- **VI, AI, VD, AD:** cálculo de las medidas características de cada cavidad, los cálculos específicos se muestran en la Tabla A2

Ventrículo izquierdo	<ul style="list-style-type: none"> <li>- Volumen en sístole y diástole</li> <li>- Fracción de eyección</li> <li>- Tensión longitudinal global</li> </ul>
Aurícula izquierda	<ul style="list-style-type: none"> <li>- Área en sístole y diástole</li> <li>- Cambio de área fraccional</li> </ul>
Ventrículo derecho	<ul style="list-style-type: none"> <li>- Área en sístole y diástole</li> <li>- Cambio de área fraccional</li> <li>- TAPSE</li> <li>- Tensión sistólica</li> </ul>
Aurícula derecha	<ul style="list-style-type: none"> <li>- Área en sístole y diástole</li> <li>- Cambio de área fraccional</li> <li>- Volumen en sístole y diástole</li> </ul>

## Tabla A2. Mediciones realizadas para cada una de las cavidades en ultrasonidos en vista de cuatro cámaras.

El módulo de Cardiac Tool tiene la opción de generar anotaciones de manera automática haciendo uso de algoritmos de Deep Learning. En este módulo, las anotaciones se crean en un solo cuadro de imagen que corresponde a la fase de sístole o de diástole, dependiendo de cuál sea seleccionada. El procedimiento para crear estas anotaciones es el siguiente:

1. Seleccionar el botón Annotate
2. Seleccionar la máscara sobre la cuál desea realizar las anotaciones
3. Seleccionar entre los botones de sístole y diástole aquel en el que desee realizar las anotaciones
4. Dar click en el botón Auto Annotation

En la Figura A10 se muestra el cálculo de las mediciones realizadas en cada una de las cavidades anotadas usando la opción de Auto Annotation, en el ultrasonido en vista de las cuatro cámaras.

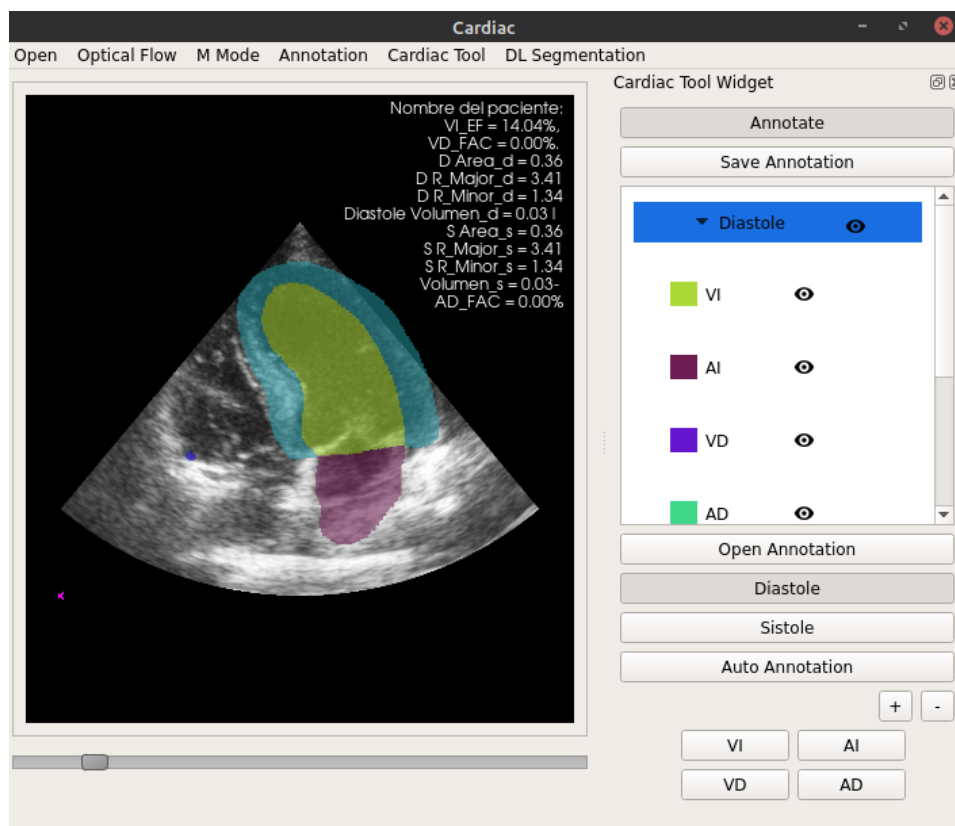
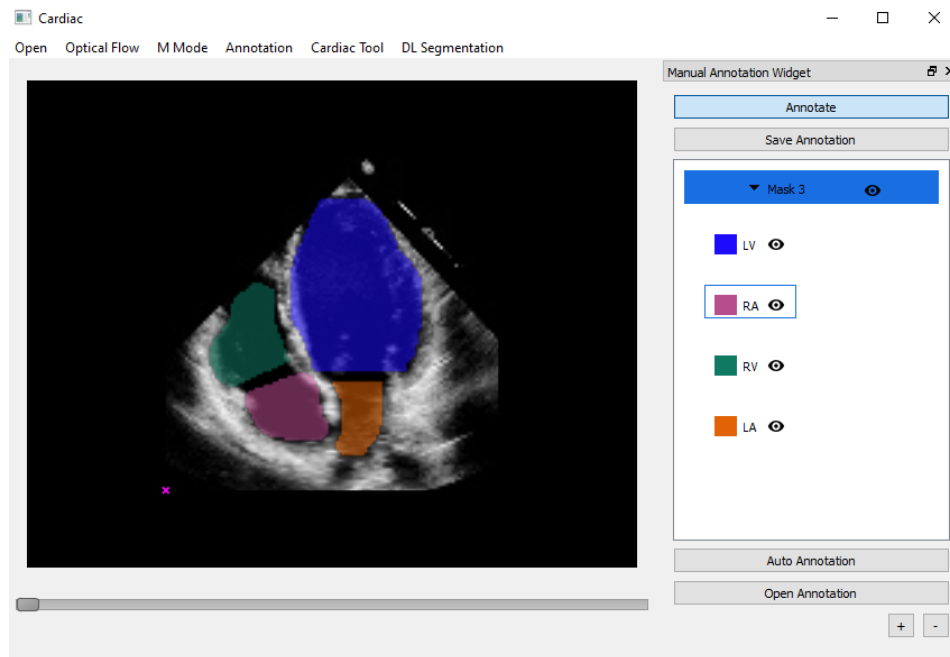


Figura A10. Cálculos realizados en las mediciones de cada una de las cavidades.

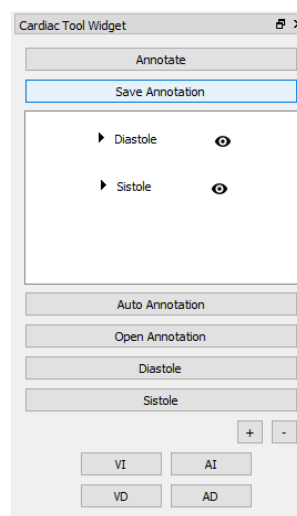


En la Figura A11 se muestra la anotación de los ventrículos y las aurículas en una imagen de ultrasonido de cuatro cámaras.



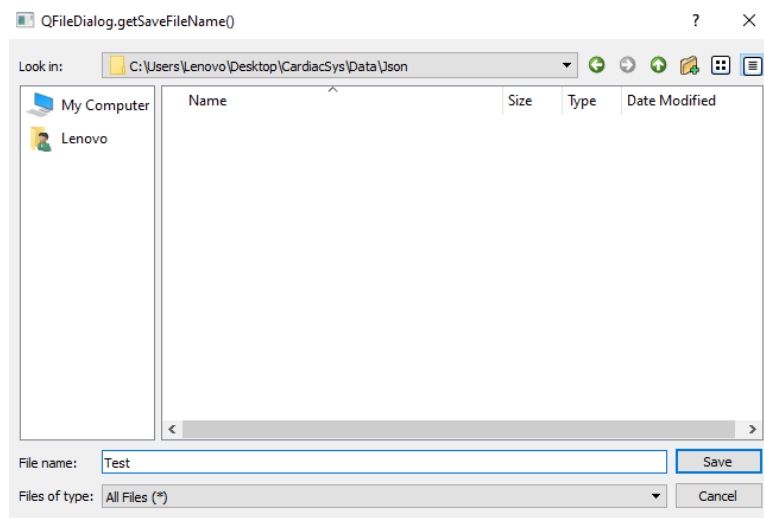
**Figura A11. Anotación de las 4 cuatro cámaras cardiacas por un experto haciendo uso del módulo de anotaciones manuales.**

Una vez creadas las máscaras y anotaciones requeridas por el usuario en una imagen o video; este puede decidir guardar las anotaciones en cualquier momento al presionar el botón "Save Annotation" en las secciones de "Manual Annotation" y Cardiac Tool"



### **Figura A12. Selección del botón “Save Annotation” en el módulo Cardiac Tool**

Al presionar el botón, la aplicación abrirá una ventana en donde el usuario podrá seleccionar el lugar en donde se crearán los archivos necesarios para guardar las anotaciones. Adicionalmente, el usuario también necesitará nombrar un archivo, y dicho nombre se utilizará como base para la creación del resto.



### **Figura A13. Ventana para seleccionar la ubicación y el nombre del archivo a guardar**

Después de haber seleccionado el lugar y el nombre del archivo donde se guardarán las anotaciones, el programa utilizará cierta información para la creación de los archivos. Se requiere una lista de las máscaras creadas ordenadas por frame y el nombre del archivo donde se dibujaron las anotaciones.

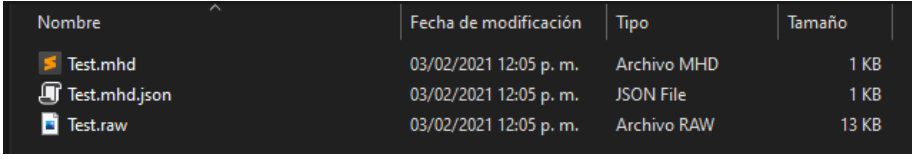
Si se están salvando anotaciones de una imagen, se creará un archivo en formato tiff por cada máscara. El nombre que tomará cada archivo se encuentra en el siguiente formato, primero el nombre que el usuario definió en los pasos anteriores, seguido por "\_", más el nombre de la respectiva máscara, un guión bajo adicional y el texto "mask".

Si en cambio, las anotaciones salvadas son de un video o formato sobre el tiempo (por ejemplo, mhd), se crearán un archivo en mhd y su respectivo raw para guardar las anotaciones. El nombre que tendrá dicho archivo

solo será el nombre que el usuario ingresó previamente seguido del respectivo formato (.mhd o .raw).

Adicionalmente, en ambas situaciones (tanto en imágenes como en videos), se creará un archivo en formato json, el cual será el seleccionado cuando el usuario quiera abrir las anotaciones guardadas. El archivo json contiene información como el nombre de las máscaras, posición de las máscaras, nombre de las anotaciones y la posición de dichas anotaciones, el nombre del archivo donde las máscaras están guardadas (el archivo tiff o mhd), y por último el archivo original donde se hicieron las anotaciones.

Los archivos creados por este proceso son necesarios para eventualmente poder abrir las anotaciones en otro momento, por lo que cuando el usuario decida abrir las anotaciones los archivos requieren estar juntos y el archivo json será el que se seleccione en dicho momento.



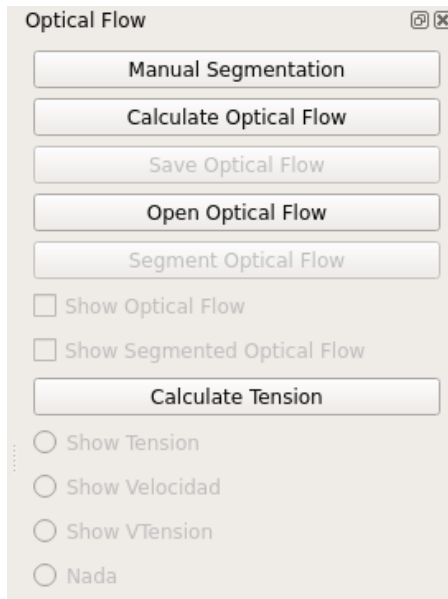
Nombre	Fecha de modificación	Tipo	Tamaño
Test.mhd	03/02/2021 12:05 p. m.	Archivo MHD	1 KB
Test.mhd.json	03/02/2021 12:05 p. m.	JSON File	1 KB
Test.raw	03/02/2021 12:05 p. m.	Archivo RAW	13 KB

**Figura A14. Archivos creados con “Save Annotation”**

### 3.4 Módulo de flujo óptico

El módulo del flujo óptico tiene como función el cálculo de los vectores de movimiento en una secuencia de imágenes de ultrasonido en modo B a través del tiempo, para la evaluación dinámica del movimiento de las estructuras cardíacas dentro de uno o más ciclos cardíacos. Esto permite al médico realizar una evaluación visual y cuantitativa del desplazamiento en diferentes estructuras cardíacas como las paredes ventriculares y las válvulas cardíacas. Este módulo permite tanto el cálculo de los vectores como la visualización e interacción con ellos. En la Figura A15 se muestra la interfaz gráfica del módulo de cálculo de flujo óptico, así como las funciones que realiza.

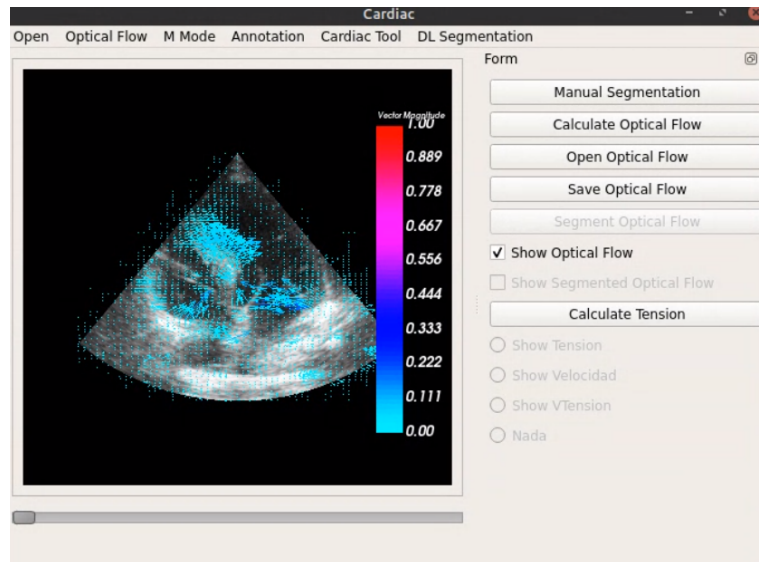
- **Manual Segmentation:** Segmentación manual de estructuras
- **Calculate Optical Flow:** Realiza el cálculo de los vectores de movimiento en una secuencia de tiempo.
- **Save Optical Flow:** Salva el resultado de los vectores de movimiento del flujo óptico como un archivo \*.npz
- **Open Optical Flow:** Abre flujos ópticos previamente salvados como un archivo \*.npz.
- **Segment Optical Flow:** obtener vectores de las segmentaciones manuales.
- **Show Optical Flow:** muestra los vectores de movimiento sobre la imagen.
- **Show Segmented Optical Flow:** muestra los vectores de movimiento segmentados sobre la imagen.
- **Calculate tension:** realiza el cálculo de las tensiones longitudinales para inhabilitar los botones radiales
- **Show tensión, Show velocidad, Show VTension, Nada:** botones radiales que muestran el valor seleccionado



**Figura A15. Optical Flow widget**

En la Figura A16 se muestran los vectores de movimiento proyectados sobre la imagen mediante una escala de color, la cual representa la magnitud del vector. El botón de “Save Optical Flow” y el checkbox “Show Optical Flow” se habilitarán una vez se haya realizado el cálculo de los vectores de flujo óptico. De igual manera, los

botones radiales de “Show Tension”, “Show Velocidad”, “Show VTension” y “Nada” se habilitarán una vez que se haya realizado el cálculo de la tensión.



**Figura A16. Vectores de movimiento proyectados sobre una imagen de ultrasonido**

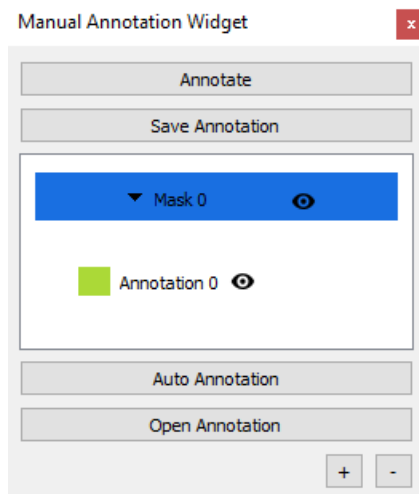
La herramienta Optical Flow permite la visualización de los vectores tangenciales de flujo óptico sobre un contorno anotado manualmente por el usuario. Para habilitar el botón de “Segment Optical Flow” es necesario contar antes con anotaciones sobre las cuales se llevará a cabo la segmentación de los vectores de flujo óptico, al obtener estos vectores segmentados se habilitará el checkbox “Show Segmented Optical Flow”.

Cuando se requiera hacer anotaciones para la segmentación de los vectores de flujo óptico, se podrán emplear en conjunto los widgets de Anotación y de Flujo Óptico, para esto será necesario abrir primero el widget de anotación (Annotation) y posteriormente abrir el widget de Flujo Óptico (Optical Flow). Una vez hecho esto, solo será necesario activar el botón “Manual Segmentation” del widget de Flujo Óptico, y cualquier anotación que se realice podrá ser empleada por ambos widgets, tanto para la generación de las máscaras del área segmentada, como para la segmentación de los vectores de flujo óptico.

En este módulo podemos realizar anotaciones de manera automática en todos los cuadros de imagen del ultrasonido que hayamos seleccionado. Para esto

es necesario utilizar en conjunto los módulos de anotación y de flujo óptico. Al dar click en el botón Auto Annotation en el módulo de anotación, se crearán las anotaciones de cada uno de los cuadros de imagen.

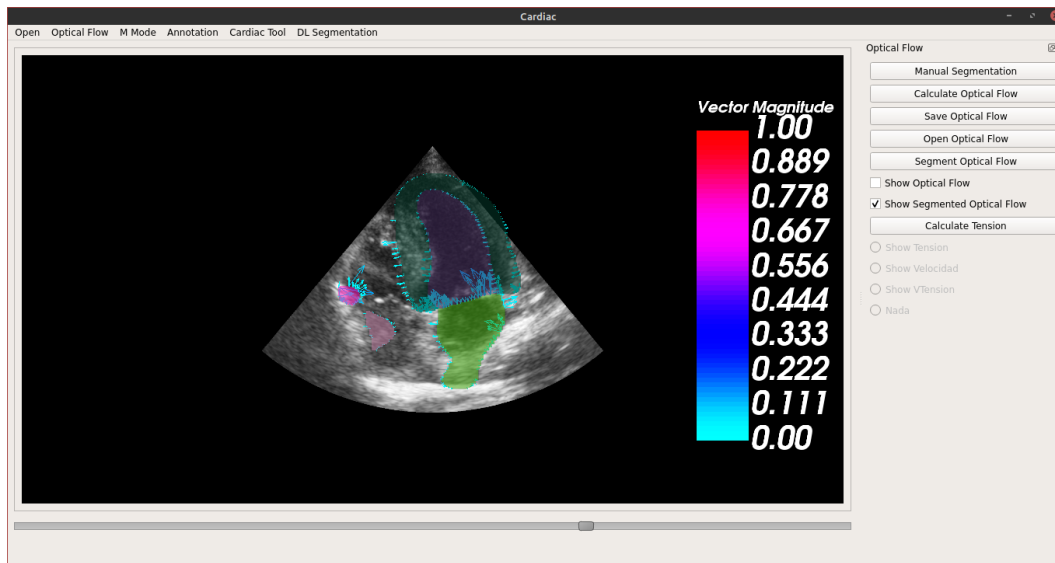
En la Figura A17 vemos el módulo de anotación, en este caso es posible usar de manera simultánea el módulo de flujo óptico y el de anotación.



**Figura A17. Manual annotation widget.**

- **Anotate:** Habilita la herramienta de segmentación manual
- **Save Annotation:** Salva la anotación manual.
- **Masks:** Máscaras de anotación realizadas sobre la imagen, se pueden realizar varias máscaras de segmentación sobre cada imagen. Este permite a su vez la eliminación de anotaciones incorrectas.
- **Auto annotation:** segmenta de manera automática las cavidades de la imagen
- **Open annotation:** permite importar o abrir anotaciones con sus respectivas máscaras de archivos que fueron previamente guardados

En la Figura A18 se muestra el uso en conjunto del widget de anotación y el widget de flujo óptico, dando un resultado donde se muestra tanto la máscara de la región donde se hizo la anotación y los vectores de flujo óptico sobre el contorno de esta región.

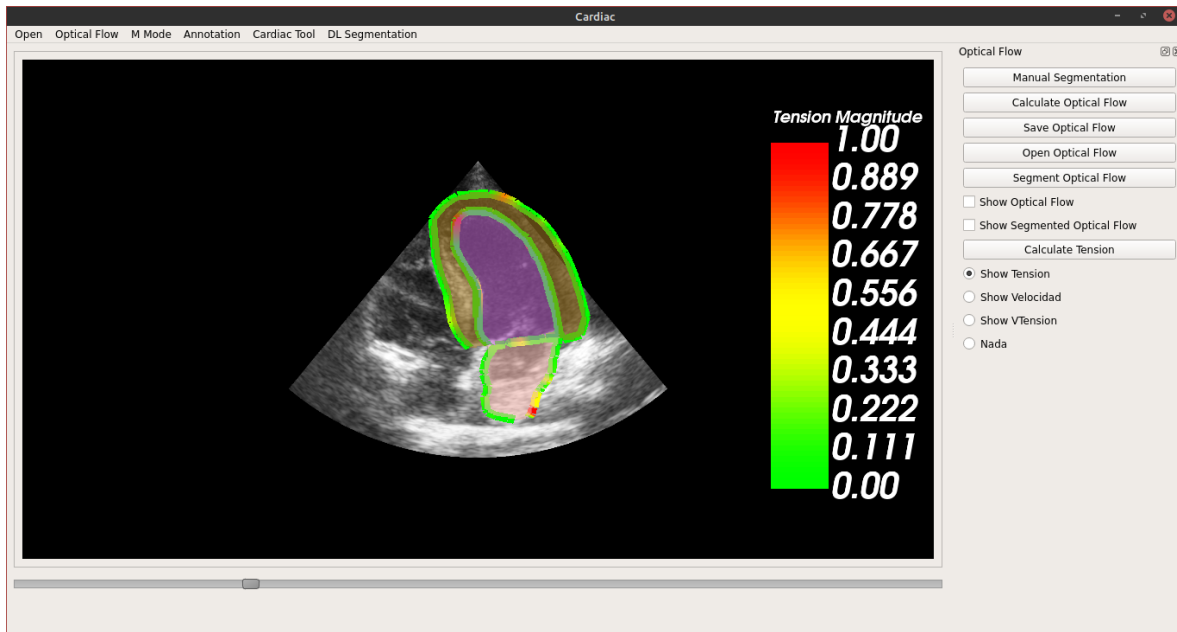


**Figura A18. Segmentación de vectores de flujo óptico y máscara de la anotación.**

En la Figura A19 se muestra la medición de la tensión y mediciones derivadas de esta en contornos marcados manualmente por un médico.

El procedimiento para los cálculos de tensión es el siguiente:

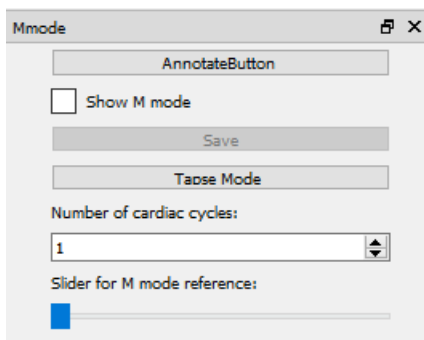
1. Ir al primer cuadro de imagen que se quiere analizar, ahí realizar las anotaciones de las cavidades a analizar
2. Pasar al siguiente cuadro de imagen y realizar las anotaciones de las mismas cavidades
3. Repetir el paso 2 hasta haber anotado todos los cuadros de imagen de la secuencia que se desean analizar
4. Dar click en el botón “Open Optical Flow”, se abrirá una ventana, seleccionar el archivo of.npz
5. Dar click en el botón “Calculate Tension”
6. Recorrer los botones radiales para seleccionar la medición que se quiere mostrar, y navegar por la secuencia para ver los resultados en los diferentes cuadros



**Figura A19. Segmentación de vectores de flujo óptico y máscara de la anotación**

### 3.5 Modulo modo M

Este módulo permite la generación de imágenes en modo M a partir de una imagen en modo B en movimiento (secuencia de tiempo), las cuales permiten visualizar características de movilidad de diferentes zonas del corazón atravesadas por un haz, esto ayuda a los médicos a evaluar el movimiento de las estructuras cardiacas como la raíz aórtica, válvula tricúspide y la válvula mitral. A su vez, de esta modalidad de imagen se pueden derivar otras medidas como el TAPSE, la cual sirve para la valoración de la función ventricular derecha en un ecocardiograma de 4 cámaras. En la Figura A20 se muestra la interfaz gráfica del módulo de modo M, así como las funciones que realiza.



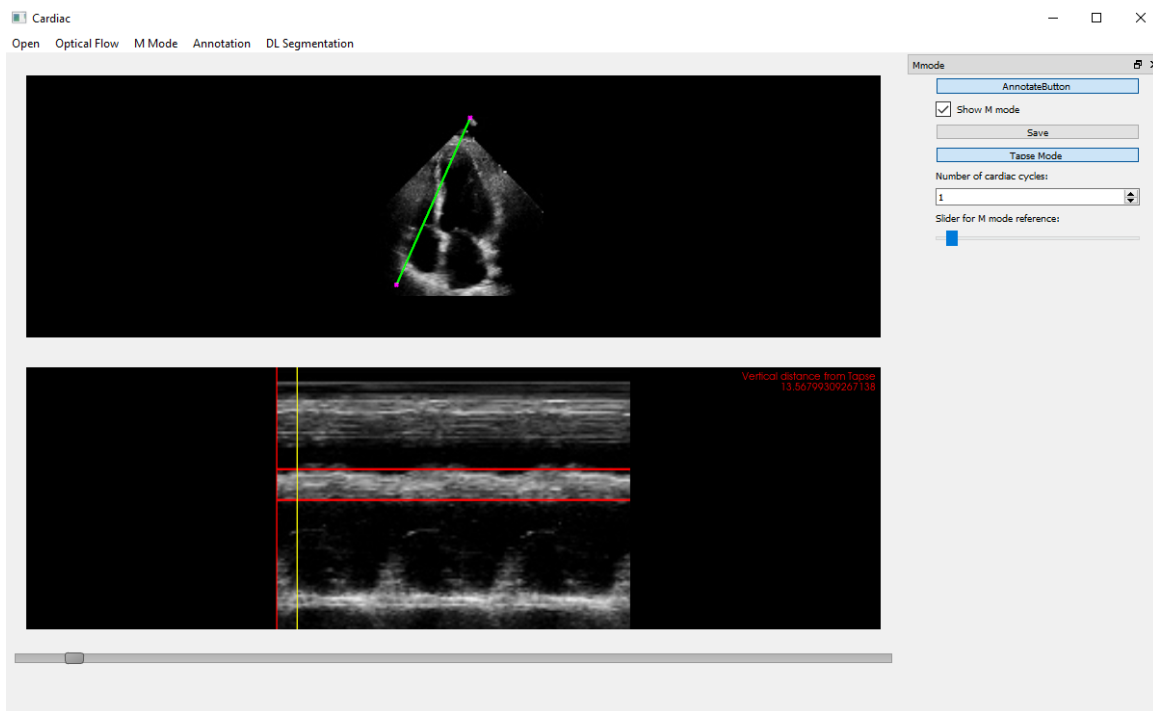
**Figura A20. Mmode widget**

- **AnotateButton:** Realiza el marcado del haz para el cálculo del modo M.
- **Show M mode:** Despliega la imagen de modo M calculada.
- **Save:** Salva la imagen modo M en formato \*.bmp
- **Tapse:** Habilita y realiza el cálculo del Tapse sobre el modo M.



- **Number of cardiac cycles:** Número de ciclos cardíacos que se quieren visualizar
- **Slider Move M mode References:** mueve referencia sobre el modo m con respecto al tiempo

En la Figura A21 se muestra la imagen de modo M calculada a partir de una secuencia de tiempo en modo B a través de un haz marcado manualmente. A su vez en esta imagen se muestra el cálculo del TAPSE sobre la imagen de modo M.



**Figura A21. Cálculo de imagen modo M y Tapse sobre una secuencia de tiempo en modo B.**

#### 4. Ejecución de la herramienta en computadoras personales.

Con el fin de poder hacer uso de la herramienta fuera del ambiente de programación, se procedió a la generación de archivos ejecutables, los cuales permiten a la herramienta correr de manera autónoma. Para realizar eso se hizo uso de la herramienta PyInstaller, la cual permite empaquetar aplicaciones de Python en diversos sistemas operativos como Windows, GNU/Linux, MacOS X, FreeBSD, Solaris y AIX. Debido a que la generación de ejecutables para cada sistema

operativo requiere que este se cree en dicho sistema, se han creado y probado los ejecutables para los siguientes sistemas operativos:

Sistema	Versión
Windows	10
MacOS X	10.15 Catalina, 10.14 Mojave
Ubuntu	20.04.1

## 5. Pruebas de funcionamiento del sistema de monitoreo cardiaco.

A continuación, se enlistan los pasos correspondientes para probar el correcto funcionamiento de la herramienta.

1. Ejecutar el sistema: Dar doble click al archivo Cardiac.exe para ejecutar la herramienta. Este proceso cargará las librerías necesarias para la ejecución de las herramientas implementadas, este proceso puede tardar varios minutos.

2. Abrir una imagen:

2.1. Dar click en el botón Open del menú principal, ver Figura A1.

2.2. Abrir el tipo de imagen que se desea visualizar, ver Sección 3.1 de este anexo.

2.3. Visualizar imagen, en este punto se debería de visualizar la imagen en el módulo de visualización, ver Figura A1.

2.4. Si se eligió abrir múltiples imágenes 2D o 2D+T, moviendo el slider que se muestra en la Figura A1 se puede navegar a través de las imágenes o los cuadros temporales del video de ultrasonido.

2.5. Dependiendo del tipo de imagen abierta se activarán los siguientes menús.

- Imágenes 2D: Se activará el menú de Annotation.

- Imágenes 2D+T: Se activarán los menús MMode, OpticalFlow, Cardiac y Annotation.

### 3. Módulo MMode.

3.1. Dar click en el botón MMode del menú principal y dar click en Compute MMode, esto abrirá el widget de la herramienta MMode, ver Figura A20.

3.2. Anotar trazó de la línea de cálculo del ultrasonido en modo M. Se aprieta el botón Annotate Button para activar la anotación manual sobre la imagen desplegada. Para realizar la anotación se da click al tercer botón del ratón para iniciar el trazo, posteriormente se arrastra el ratón hasta el punto final del trazo para el cálculo del MMode. Finalmente se vuelve aprieta ctrl+click en el tercer botón del mouse para finalizar el trazo, ver Figura A21.

3.3. Dar click en Show MMode para visualizar la imagen de ultrasonido de modo M procesada a partir del trazo manual realizado en el paso 3.2.

3.4. Siguiendo se aprieta el botón TAPSE Mode para realizar el cálculo del TAPSE. Esto activará la interacción con el visualizador del ultrasonido de modo M. Para calcular el TAPSE se debe de dar click en el punto mínimo del movimiento de la válvula Tricúspide, seguido por un click en la parte máxima del movimiento, ver Figura A21.

3.5. Cerrar módulo, esto deberá de eliminar el visualizador del modo M así como el trazó sobre la imagen.

### 4 Annotation + OpticalFlow.

4.1. Dar click en el botón Annotation del menú principal y dar click en Compute Manual, esto abrirá el widget de la herramienta de anotación, ver Figura A17.

4.2 Dar click en el botón OpticalFlow del menú principal y dar click en Compute Opticalflow, esto abrirá el widget de la herramienta OpticalFlow, ver Figura A15.

4.3. Segmentar las imágenes. Dar click en el botón de Manual Segmentation del widget de OpticalFlow, esto indicará que se realizará una segmentación a través del widget de anotación.

Posteriormente se realizará la segmentación automática utilizando el botón de Automatic Segmentation de la herramienta de anotación. Esto realizará la segmentación mediante redes de aprendizaje profundo de todos los cuadros del ultrasonido 2D+T o en todas las imágenes si se abrieron múltiples imágenes 2D, este proceso puede tardar varios minutos dependiendo del número de cuadros o imágenes a procesar. Al finalizar se visualiza una máscara de segmentación sobre cada una de las cavidades cardíacas de todos los cuadros o imágenes.

4.4. Cálculo del flujo óptico. Esto se realiza al apretar el botón de Calculate opticalflow, este proceso puede tardar varios minutos dependiendo del número de cuadros o imágenes a procesar.

4.5. Visualizar flujo óptico. Esto se realiza escogiendo una de las siguientes opciones: a) show Optical Flow, muestro el flujo óptico sobre toda la imagen; b) muestra el flujo óptico sobre los bordes de las segmentaciones realizadas.

4.6. Calcular parámetros de velocidad y tensión de las paredes cardíacas. Esto se realiza mediante el botón de Calculate Tensions. Al finalizar el proceso se activarán los botones radiales para seleccionar el parámetro que se desea visualizar sobre la imagen.

4.7. Se cierra la herramienta. Esto debería de borrar las anotaciones y los vectores de flujo óptico sobre la imagen.

## 5. Cardiac

5.1. Dar click en el botón Cardiac del menú principal y dar click en Cardiac Tool, esto abrirá el widget de la herramienta de cálculo de índices cardíacos, ver Figura A9.

5.2. Se navega en la imagen 2D+T o múltiples imágenes hasta encontrar un cuadro de Diástole y se segmenta automáticamente mediante el botón Auto Segmentation.

5.3. Se navega en la imagen 2D+T o múltiples imágenes hasta encontrar un cuadro de Sístole y se segmenta automáticamente mediante el botón Auto Segmentation.

5.4. Se calculan los índices morfológicos de las cavidades cardiacas apretando los botones correspondientes: ventrículo izquierdo (VI), ventrículo derecho (VD), aurícula izquierda (AI) y aurícula derecha (AD). Los parámetros calculados deberán de aparecer en la esquina superior derecha de la ventana de visualización.

5.5. Cerrar widget de la herramienta cardiaca. Esto deberá eliminar las anotaciones y los índices desplegados sobre la imagen.

6. Cerrar herramienta cardiaca