

**GENERADOR DE RUIDOS CARDIACOS**

SANDRO FABIAN CASTRO GUTIERREZ

FUNDACION UNIVERSITARIA LOS LIBERTADORES  
FACULTAD DE INGENIERÍA Y CIENCIAS BÁSICAS  
BOGOTÁ D.C.

2021

# GENERADOR DE RUIDOS CARDIACOS

SANDRO FABIAN CASTRO GUTIERREZ

Trabajo de grado presentado para optar al título de Ingeniero  
Electrónico

Director

LUIS FERNANDO RÍOS SENDOYA

Ingeniero Electrónico, MSc I&C

FUNDACION UNIVERSITARIA LOS LIBERTADORES  
FACULTAD DE INGENIERÍA Y CIENCIAS BÁSICAS  
BOGOTÁ D.C.

2021

NOTA DE ACEPTACIÓN

---

---

---

---

---

Firma del Presidente del  
Jurado

---

\_\_\_\_\_  
Firma del Jurado

---

Firma del Jurado

Bogotá D.C., Septiembre de 2021

---

## DEDICATORIA

Dedico este trabajo a mi Familia y amigos por ayudarme a continuar frente a lo duro de las circunstancias.

**Sandro Fabián Castro  
Gutiérrez**

## **AGRADECIMIENTOS**

Agradezco a todas las personas que me apoyaron en este arduo camino , gracias a ellos se puede evidenciar el éxito de este trabajo que viene siendo desarrollado desde el año 2018

## Contenido

<b>RESUMEN GENERAL DEL TRABAJO DE GRADO .....</b>	<b>18</b>
<b>RESUMEN .....</b>	<b>18</b>
<b>GENERAL SUMMARY OF WORK OF GRADE .....</b>	<b>12</b>
<b>ABSTRACT .....</b>	<b>12</b>
<b>1. PROBLEMA .....</b>	<b>19</b>
<b>1.1 PLANTEAMIENTO DEL PROBLEMA.....</b>	<b>19</b>
<b>1.2 FORMULACIÓN DEL PROBLEMA .....</b>	<b>19</b>
<b>2.1 OBJETIVOS GENERALES .....</b>	<b>19</b>
<b>2.2 OBJETIVOS ESPECIFICOS .....</b>	<b>19</b>
<b>3. JUSTIFICACIÓN.....</b>	<b>19</b>
<b>4. DELIMITACIÓN .....</b>	<b>20</b>
<b>5. INTRODUCCIÓN .....</b>	<b>20</b>
<b>6. CONCEPTOS DE DESARROLLO DE SOFTWARE.....</b>	<b>21</b>
<b>6.1 PARADIGMAS DE DESARROLLO. ....</b>	<b>21</b>
<b>6.1.1 PROGRAMACIÓN IMPERATIVA .....</b>	<b>21</b>
<b>6.1.2 PROGRAMACIÓN FUNCIONAL .....</b>	<b>22</b>
<b>6.1.3 PROGRAMACIÓN ORIENTADA A OBJETOS.....</b>	<b>23</b>
<b>6.1.3.1 ENCAPSULACIÓN.....</b>	<b>24</b>
<b>6.1.3.2 HERENCIA .....</b>	<b>25</b>
<b>6.1.3.3 POLIMORFISMO .....</b>	<b>26</b>
<b>6.1.4 PROGRAMACIÓN LÓGICA .....</b>	<b>27</b>
<b>7. CASOS DE USO .....</b>	<b>28</b>
<b>7.1 ESTRUCTURA GRÁFICA DE UN CASO DE USO .....</b>	<b>28</b>
<b>7.2 ESTRUCTURA ESCRITA DE UN CASO DE USO. ....</b>	<b>29</b>
<b>8. DIAGRAMA DE SECUENCIA.....</b>	<b>29</b>
<b>9. DIAGRAMA DE CLASES.....</b>	<b>30</b>
<b>10. EL CORAZÓN.....</b>	<b>31</b>

<b>10.1 VALVULAS DEL CORAZÓN.....</b>	<b>33</b>
<b>10.2 AURICULAS.....</b>	<b>34</b>
<b>10.3 VENTRICULAS .....</b>	<b>34</b>
<b>10.4 ELECTROCARDIOGRAMA .....</b>	<b>36</b>
<b>10.5 FONOCARDIOGRAMA .....</b>	<b>37</b>
<b>11. SIMULACIÓN GRÁFICA DEL CORAZÓN.....</b>	<b>37</b>
<b>12. CONCLUSIONES .....</b>	<b>44</b>
<b>13. BIBLIOGRAFIA .....</b>	<b>45</b>

## LISTA DE ILUSTRACIONES

Ilustración 1. Fragmento de código Cobol. Fuente(autor) .....	22
Ilustración 2. Fragmento de código imperativo. fuente(autor).....	23
Ilustración 3. Fragmento de código funcional.fuente(autor) .....	23
Ilustración 4. Clase Carro POO. fuente(autor) .....	24
Ilustración 5. Instanciación de un objeto en c#. fuente(autor) .....	24
Ilustración 6. Encapsulacion mediante clases.....	25
Ilustración 7. Clase perro heredará atributos de la clase animal .....	25
Ilustración 8. Establecimiento de la herencia de un atributo.....	26
Ilustración 9. Establecimiento de una variable en común que tendrá la clase carro y moto heredando características de la clase vehículo. fuente (autor). .....	26
Ilustración 10. Polimorfismo establecido en el atributo nombre. fuente(autor) .....	27
Ilustración 11. Diagrama de clase persona.Fuente(personal) .....	31
Ilustración 12. Flujo general de simulación de movimientos cardiacos.Fuente(autor).....	42
Ilustración 13. Diagrama de clases de la clase clsCanvas para la simulación.fuente(autor) .....	43
Ilustración 14. Diagrama de caso de uso para el manejo de la clase clsCanvas.fuente(autor) .....	43



## LISTA DE FIGURAS

Figura 1.Caso de uso diagrama grafica .fuente libro Ian Sommerville_ Victor Campos Olguin - Ingeniería de software-Pearson Educación (2011), pág 107 .....	28
Figura 2.Diagrama de secuencia tomado de Ian Sommerville_ Victor Campos Olguin - Ingeniería de software-Pearson Educación (2011). pag 127 .....	30
Figura 3, Corazón humano , tomado del libro fundamentos de medicina, cardiología , sexta edición , portada.....	33
Figura 4. Flujo de la sangre parte 1.tomado de el aparato circulatorio, por conrad j. Storad pag 22 .....	35
Figura 5.Flujo sanguíneo parte 2 y 3 . tomado de el aparato circulatorio, por conrad j. Storad pag 23 .....	35
Figura 6. Ciclo cardiaco ,tomado del libro Cardiología de Jose Fernando Guadalajara sexta edición, pag 43 .....	36
Figura 7. Conjunto ondas que conforman el electrocardiograma , tomado del libro Cardiología de Jose Fernando Guadalajara sexta edición, pág. 75.....	37
Figura 8, Partes Básicas del corazón. tomado de la web <a href="https://psicologiymente.com/salud/partes-del-corazon">https://psicologiymente.com/salud/partes-del-corazon</a> .....	40

## LISTA DE ANEXOS

ANEXO CÓDIGO NODE.JS .....	46
ANEXO CÓDIGO DE SIMULACIÓN GRÁFICA HTML - JAVASCRIPT .....	47
ANEXO CÓDIGO DE SIMULACIÓN GRÁFICA ARDUINO .....	52

## GENERADOR DE RUIDOS CARDIACOS

## RESUMEN GENERAL DEL TRABAJO DE GRADO

TITULO	:	GENERADOR DE RUIDOS CARDIACOS
AUTORES	:	SANDRO FABIAN CASTRO GUTIERREZ
FACULTAD	:	INGENIERÍA
PROGRAMA	:	INGENIERÍA ELECTRÓNICA
DIRECTOR TRABAJO DE GRADO	:	LUIS FERNANDO RÍOS SENDOYA

### RESUMEN

Esta investigación muestra las herramientas necesarias para la simulación de comportamientos cardiacos mediante elementos gráficos significativos, que ayudará a interesados en el campo de la ingeniería y medicina al análisis de las actividades cardiacas.

La implementación de desarrollo de software dará los elementos necesarios para el gestionamiento del modelado antes mencionado, programación que se encargará de emular el comportamiento del corazón humano.

#### **Palabras claves:**

Arduino, NodeJs, JavaScript, Html5 .

**GENERAL SUMMARY OF WORK OF GRADE**

TITLE : GENERADOR DE RUIDOS CARDIACOS

AUTHOR(S) : SANDRO FABIAN CASTRO GUTIERREZ

FACULTY : INGENIERÍA

PROGRAM : INGENIERÍA ELECTRÓNICA

DIRECTOR : LUIS FERNANDO RÍOS SENDOYA

**ABSTRACT**

This research shows the necessary tools for the simulation of cardiac behaviors through significant graphic elements, which will help those interested in the field of engineering and medicine to analyze cardiac activities.

The software development implementation will provide the necessary elements to manage the aforementioned modeling, programming that will be responsible for emulating the behavior of the human heart.

**Keywords:**

Arduino, NodeJs, JavaScript, Html5.

## **1. PROBLEMA**

### **1.1 PLANTEAMIENTO DEL PROBLEMA**

Dentro de la Fundación Jorge Reynolds se buscará la implementación de un sistema de Modelado Cardíaco, herramientas que permitan recrear el comportamiento de sístole, diástole y pulso, de esa manera se podrá entender otros fenómenos del corazón.

### **1.2 FORMULACIÓN DEL PROBLEMA**

¿Cómo el diseño de software y el modelamiento gráfico induce a la comprensión de los Movimientos Cardíacos?

## **2. OBJETIVOS**

### **2.1 OBJETIVOS GENERALES**

Diseñar un conjunto de herramientas que permitan dilucidar el comportamiento del corazón humano mediante la modelación gráfica.

### **2.2 OBJETIVOS ESPECIFICOS**

Crear un software que permita graficar el corazón humano y dentro de este generar actividades cardíacas en función de la información de una señal provista por microcontroladores en este caso usando como herramientas Arduino y motores de Desarrollo como NodeJs.

## **3. JUSTIFICACIÓN**

En la actualidad en diferentes instituciones, es indispensable la implementación de investigación aplicada, utilizar modelación y simulación de sistemas haciendo uso de herramientas computacionales, por ende se entiende que en la fundación Jorge Reynolds se necesite la creación de modelos para el análisis del comportamiento del cuerpo humano, puntualmente del músculo cardíaco, con esos

resultados los expertos en el tema( Médicos, Ingenieros, Enfermeros) podrán entender, analizar y gestionar soluciones ante diferentes problemas del corazón que tenga un ser humano.

#### 4. DELIMITACIÓN

El Área de interés del proyecto es procurar la implementación de herramientas para que el cuerpo médico y científico de diversas entidades puedan entender el comportamiento del corazón.

#### 5. INTRODUCCIÓN

En la actualidad es difícil pensar el mundo sin diversas aplicaciones informáticas y electrónicas, se evidencian en distintos aspectos del mundo contemporáneo, al hacer uso de un celular, investigar información puntual en un café internet, cosas asociadas a una constante cotidianidad día a día y que su complejidad y asombro pasan desapercibidos. No lo es tanto cuando se escucha en las noticias la nueva creación de una nave no tripulada que surca el espacio o una nueva tecnología simulando el comportamiento de un sistema industrial en una importante compañía, eventos tan complejos como el modelamiento de procesos naturales de un arrecife.

Se evidencia entonces, el valor estratégico que tiene en la época actual de varios mecanismos computacionales y electrónicos para lograr diferentes tareas de fundamental importancia a el ser humano, mecanismos como el software que permiten entre varias tareas la simulación y modelación de sistemas de distinta naturaleza y gama, tales como los sistemas fisiológicos humanos.

Pero: ¿Cómo funciona el Software, cuya naturaleza permite lograr todas estas magnificas innovaciones?

¿Cuáles son los fundamentos que permiten la Simulación y Modulación de sistemas como los del cuerpo humano, por tomar un elemento de este, el Corazón?

## 6. CONCEPTOS DE DESARROLLO DE SOFTWARE

Se entiende Programa por un conjunto de instrucciones entregadas a sistemas de cómputo para ser interpretadas, obteniendo una respuesta específica, plasmada en diversos periféricos, por ejemplo, una pantalla, por ende, se va a analizar las herramientas de creación de un software.

Para poder participar en la implementación de técnicas en programas, se tendrá en cuenta los siguientes conceptos indispensables:

### 6.1 PARADIGMAS DE DESARROLLO.

En el momento de desarrollar software hay diferentes maneras o estilos para la creación del mismo, ese es el concepto básico para definir lo que es un paradigma en el mundo de creación de programas informáticos, al tener en cuenta esta definición se ahondara en las diferentes y más importantes tipos de paradigmas usados en la industria digital.

#### 6.1.1 PROGRAMACIÓN IMPERATIVA

Es uno de los paradigmas de programación más viejos dentro de la escena, pero con gran dominio dentro del mundo del desarrollo debido a que tiene una mayor cercanía con la arquitectura de máquina y eso hace eficiente el uso de los recursos de la misma.

Por tomar un ejemplo, en la actualidad del comercio mundial, el Core de los bancos usan software escrito en Fortran, Cobol, Basic.

Teniendo en cuenta se pasará a definir la naturaleza de este tipo de programación como un encadenamiento de instrucciones una después de otra para especificar una tarea definida usando estructuras de control o anidamientos.

A Continuación, se verá el uso de un conjunto de instrucciones que demuestran el establecimiento de esta programación imperativa en Cobol.



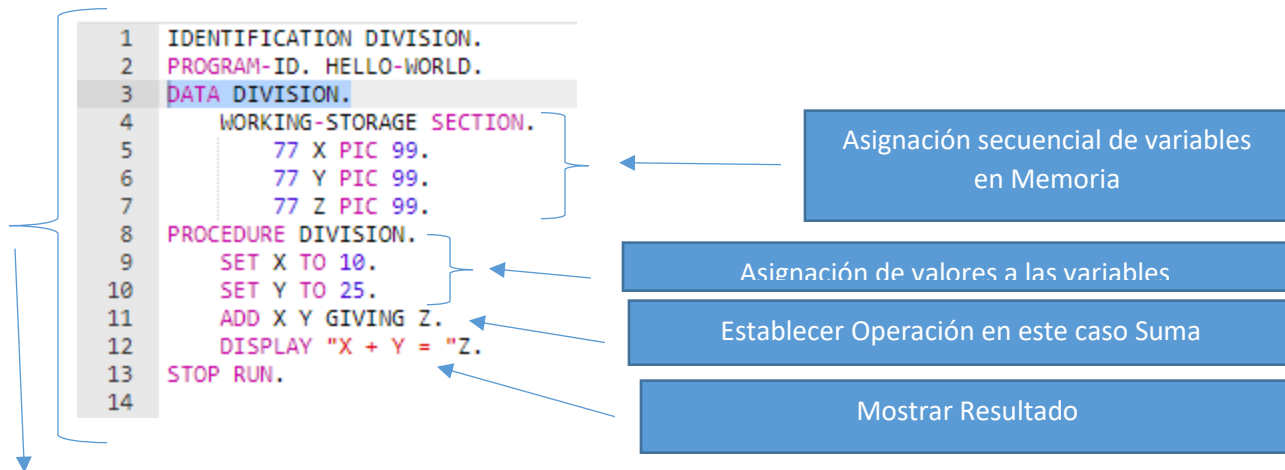


ILUSTRACIÓN 1, FRAGMENTO DE CÓDIGO COBOL. FUENTE(AUTOR)

Bloque General (Cuando se habla de División es la anidación de los distintos elementos del código)

### 6.1.2 PROGRAMACIÓN FUNCIONAL

Es un tipo de paradigma que valga la redundancia se caracteriza por la declaración de funciones que tiene como atributos el recorte de bucles y la inmutabilidad en el valor de las variables, eso que significa que desde que adquiere un valor la variable este no cambiara mientras se ejecute la programación funcional, un ejemplo clásico es el código lamda que se usa en C# y con el siguiente ejemplo se podrá ver claramente la diferencia entre este tipo de paradigma y la programación imperativa.

```

string respuesta = String.Empty;
int[] variable = new int[] { 1, 2, 3, 4, 5 };
for (int i = 0; i < variable.Count(); i++)
{
    if (variable[i] == 2) {
        respuesta = variable[i].ToString();
        break;
    }
    else {
        respuesta = "otro valor";
    }
}

```

Se establece una programación imperativa mediante un bucle for, para buscar dentro de "variable" el numero 2

ILUSTRACIÓN 2. FRAGMENTO DE CÓDIGO IMPERATIVO. FUENTE(AUTOR)

```

string respuesta = String.Empty;
int[] variable = new int[] { 1, 2, 3, 4, 5 };
respuesta = variable.Where(a => a == 2).ToArray()[0].ToString();
return respuesta;

```

ILUSTRACIÓN 3. FRAGMENTO DE CÓDIGO FUNCIONAL. FUENTE(AUTOR)

Mediante la función Where (), al buscar el numero dos dentro de "variable" se adquiere el valor sin necesidad de un bucle, se evidencia la programación funcional en este caso.

### 6.1.3 PROGRAMACIÓN ORIENTADA A OBJETOS

Las Características que usa este paradigma se basan en estructurar los datos en torno a una clase que al invocarse dentro del código manifestará un conjunto de comportamientos y atributos únicos como un objeto de la vida real.

Por tomar un ejemplo, un carro es un objeto tiene unos atributos como el color, el tamaño y una serie de comportamientos, como andar, frenar.

```
public class Carro
{
    public string Nombre;
    public string Color;
    public string Marca;
    void Frenar() { }
    void Acelerar() { }
}
```

ILUSTRACIÓN 4. CLASE CARRO POO. FUENTE(AUTOR)

Para crear un Objeto de la clase carro solo basta instanciar el elemento como por ejemplo en el lenguaje C#, que se haría de la siguiente manera.

```
Carro Objeto = new Carro();
```

ILUSTRACIÓN 5. INSTANCIACIÓN DE UN OBJETO EN C#. FUENTE(AUTOR)

Ahora se hablará de los principios de la programación orientada a objetos:

### 6.1.3.1 ENCAPSULACIÓN

Es la vinculación que existe entre los distintos comportamientos y atributos dentro de una clase y los privilegios que existen para acceder a los antes descritos, para entenderlo se mirará el siguiente ejemplo.

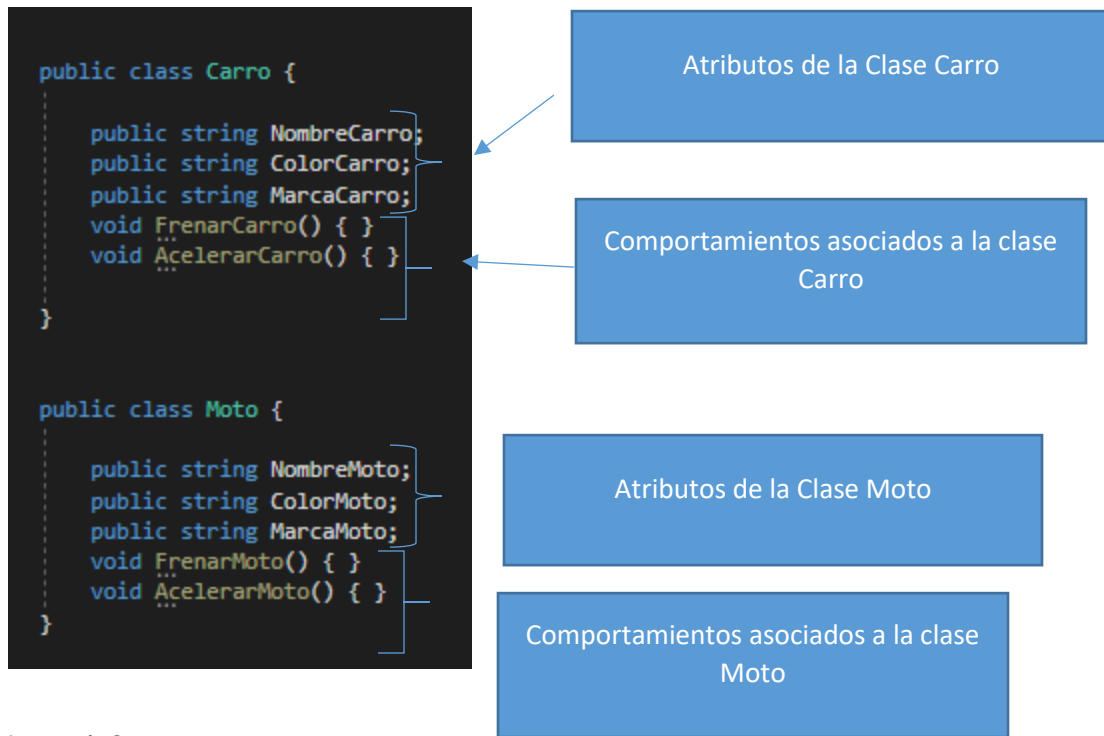


ILUSTRACIÓN 6. ENCAPSULACION MEDIANTE CLASES

En las anteriores imágenes se puede evidenciar como mediante el establecimiento de una determinada clase se encapsulan distintos métodos y atributos accesibles solo dentro de un objeto que instancie la clase.

### 6.1.3.2 HERENCIA

No es más que la capacidad de entregar los comportamientos y funcionalidades que tiene una clase A a una clase B y se verá en el siguiente ejemplo.

```

public class Animal {
    public string Nombre;
}

public class Perro : Animal {
}

```

ILUSTRACIÓN 7. CLASE PERRO HEREDARÁ ATRIBUTOS DE LA CLASE ANIMAL

Se analiza una clase de nombre Animal, esta posee un atributo denominado Nombre.

Otra clase cuyo nombre es Perro, obtiene esa misma cualidad en el momento en el que establece una herencia de atributos de la Clase Animal.

Propiamente dicho la clase Perro no tiene un atributo denominado Nombre, pero obtiene este por herencia de la clase Animal.

```
Perro Perro = new Perro();  
Perro.Nombre = "Paco";
```

ILUSTRACIÓN 8. ESTABLECIMIENTO DE LA HERENCIA DE UN ATRIBUTO

### 6.1.3.3 POLIMORFISMO

Es un conjunto de cualidades y comportamientos que dependiendo de que objeto la invoque va a tener una respectiva acción.

```
public class Vehiculo {  
    public string Nombre;  
}  
  
public class Carro : Vehiculo { }  
  
public class Moto : Vehiculo { }
```

ILUSTRACIÓN 9. ESTABLECIMIENTO DE UNA VARIABLE EN COMÚN QUE TENDRÁ LA CLASE CARRO Y MOTO HEREDANDO CARACTERÍSTICAS DE LA CLASE VEHÍCULO. FUENTE (AUTOR).

Se ve en la imagen de arriba las Clases Carro y Moto que heredan el atributo Nombre, la capacidad de polimorfismo que tiene este atributo permitirá que cada vez que se cree un objeto sea de la Clase Carro o la clase Moto pueda tener un diferente accionar como se verá en el siguiente ejemplo.

```
Carro oCarro = new Carro();  
oCarro.Nombre = "Mazda";  
  
Moto oMoto = new Moto();  
oMoto.Nombre = "Ducati";
```

Nombre	Valor
oCarro.Nombre	"Mazda"
oMoto.Nombre	"Ducati"

ILUSTRACIÓN 10. POLIMORFISMO ESTABLECIDO EN EL ATRIBUTO NOMBRE. FUENTE(AUTOR)

Se observa que a pesar de que se invoca el mismo atributo en dos clases, siendo heredada por una misma clase padre, los valores cambian, ahí se ve claramente cómo funciona el polimorfismo.

#### 6.1.4 PROGRAMACIÓN LÓGICA

Este tipo de programación se caracteriza por el uso de afirmaciones o reglas que no son más que un conjunto de condiciones o conclusiones al formular un algoritmo, como se verá en el siguiente ejemplo. Algo que se caracteriza dentro de la programación lógica es:

- Aplicaciones en el mundo de las matemáticas
- Procesamiento de herramientas para lenguaje natural
- Interacción con Base de datos

## 7. CASOS DE USO

Es una técnica de administración de requerimientos, se caracteriza por la identificación de unos actores relacionados con el proceso y su interacción con el sistema, la información relacional puede ser una descripción tácita o el uso de gráficos descriptivos. En resumidas cuentas, es un conjunto de requerimientos que definen la tarea establecida por un ejecutivo de cuenta y que un miembro del equipo de software tendrá que desarrollar.

### 7.1 ESTRUCTURA GRÁFICA DE UN CASO DE USO

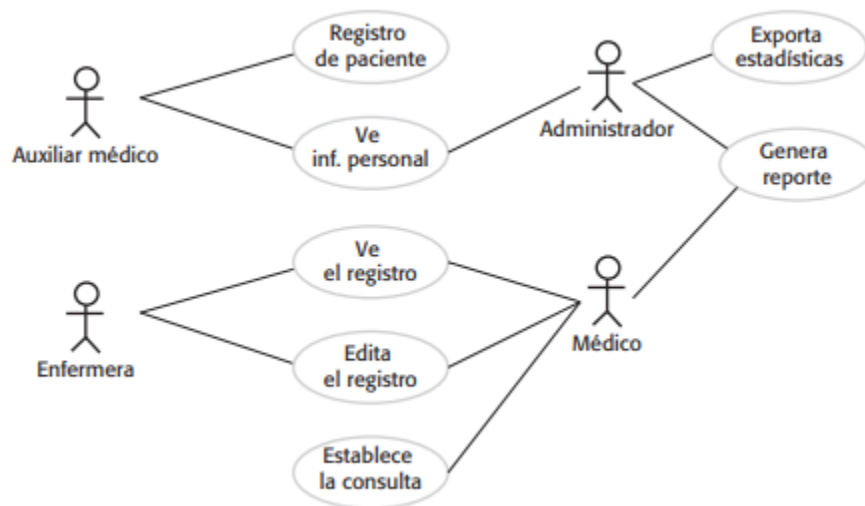


FIGURA 1. CASO DE USO DIAGRAMA GRÁFICA .FUENTE LIBRO IAN SOMMERVILLE\_ VICTOR CAMPOS OLGUIN - INGENIERÍA DE SOFTWARE-PEARSON EDUCACIÓN (2011),

Esta es la estructura de un caso de uso gráfico, se establece un actor y unos comportamientos asociados al mismo. Se percibe claramente que este presenta interacciones establecidas entre diferentes miembros de un hospital.

## 7.2 ESTRUCTURA ESCRITA DE UN CASO DE USO.

Se establece de la siguiente manera

Nombre: Nombre del caso de Uso.

Actor: Clase establecida que interactúa en el sistema.

Prerrequisito: Parámetro inicial que establecerán las bases para el comportamiento de la tarea establecida en el caso de uso.

Objetivo: El porqué de la tarea.

Postrequisito: Resultado de la ejecución de la tarea.

Regla de Negocio: Establecimiento de comportamientos necesarios y esperados del caso de uso.

Teniendo en cuenta el diagrama gráfico y los elementos para la elaboración del requisito se tendría un ejemplo tangible volviendo al caso del hospital, por ende, se usa un caso de uso que se evidencia en la figura 1, un caso de uso para las tareas del auxiliar médico.

Nombre: Ver inf Paciente.

Actor: Recepcionista Médico.

Prerrequisito: Haber registrado previamente la información del usuario.

Objetivo: obtener la información del usuario de Hospital.

Postrequisito: Visualizar la información exacta del paciente.

Reglas de negocio: validar el previo ingreso de la información del usuario.

## 8. DIAGRAMA DE SECUENCIA

Es una herramienta que modela la interacción entre los actores y objetos dentro de un sistema.



Se verá con el siguiente bosquejo.

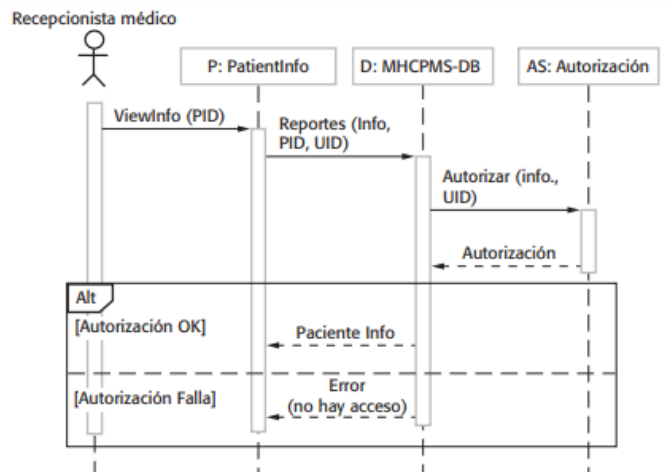


FIGURA 2. DIAGRAMA DE SECUENCIA TOMADO DE IAN SOMMERVILLE\_ VICTOR CAMPOS OLGUIN - INGENIERÍA DE SOFTWARE-PEARSON EDUCACIÓN (2011). PAG 127

## 9. DIAGRAMA DE CLASES

Anteriormente, se definió que es un Objeto, un objeto dentro del mundo del desarrollo de software es la abstracción de una entidad real que encapsula comportamientos y atributos mediante una clase, para digerirlo más fácil, se tendrá el siguiente ejemplo a colación.

Una persona tiene comportamientos y atributos

Comportamientos:

- Dormir
- Comer
- Caminar

Atributos

- Peso

- Sexo
- Edad

Si se define a una persona dentro del diagrama de clases, se tendría lo siguiente:

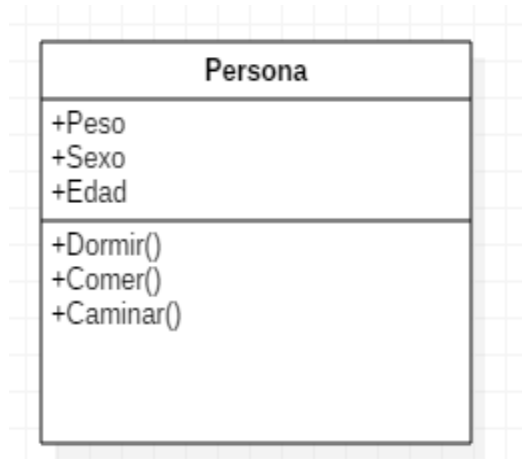


ILUSTRACIÓN 11. DIAGRAMA DE CLASE PERSONA.FUENTE(PERSONAL)

Ya entendiendo todo el engranaje con respecto a la naturaleza del software, como funciona, sus paradigmas y vías de diseño, se analizará el sistema que precisamente se desea simular en el trabajo de grado y es el corazón.

## 10. EL CORAZON

El cuerpo humano es un sistema compuesto de elementos biológicos, y electroquímicos, cuyo motor principal es el corazón, si esta falla la transmisión de oxígeno y el bombeo de sangre será imposible, para ello este genera un conjunto de señales que coordina la operación antes mencionada.

Se entiende por corazón como el principal órgano del sistema circulatorio, que está en el tórax, bombeando sangre por todo el cuerpo, se conforma por 4 cavidades, 2 superiores (auriculares) y dos inferiores (ventriculares).

Las aurículas se encargan de recibir la sangre para enviarla a los ventrículos y estos de expulsarla nuevamente para repartirla por todo el cuerpo.

Otros elementos importantes que tiene el corazón es el endocardio que es una fina capa que se encarga de cubrir las válvulas del corazón y se encarga de que la sangre fluya de manera correcta y eficiente sobre los tejidos. Gracias a esto, los elementos sanguíneos no se pegan a las paredes del corazón, si esto ocurre puede ocasionar perjuicio a los tejidos, esto también ayuda a que los tejidos del corazón se peguen cuando laten.

El miocardio se encarga de impulsar la sangre a través del corazón

El epicardio mantiene el corazón dentro del tórax además impide que el corazón se ensanche demasiado cuando aumenta el volumen sanguíneo y regula los movimientos del musculo cardiaco y lo protege contra infecciones.



FIGURA 3, CORAZÓN HUMANO , TOMADO DEL LIBRO FUNDAMENTOS DE MEDICINA, CARDIOLOGÍA , SEXTA EDICIÓN , PORTADA.

Teniendo en cuenta como está conformado el corazón se podrá entender en un determinado momento como el comportamiento que él tiene, genera una serie de señales, que indican su funcionamiento y su estado.

Por ende, es indispensable profundizar en ciertas partes y actividades del corazón.

### 10.1 VALVULAS DEL CORAZÓN

Dentro del corazón aparecen 4 válvulas específicas:

Válvula tricúspide. Esta válvula se delimita entre el ventrículo derecho y la aurícula derecha y se encarga de la administración del flujo sanguíneo entre estas dos cámaras.

Válvula Mitral. Esta Válvula se delimita entre el ventrículo izquierdo y la aurícula izquierda, su tarea es el viaje de la sangre cargada de oxígeno por los pulmones pase de la aurícula al ventrículo.

Válvula Pulmonar. Administra el paso de la sangre desde el ventrículo derecho a las arterias pulmonares para que el pulmón valga la redundancia oxigene el flujo sanguíneo.

Válvula Aórtica. Es la compuerta que permite el flujo sanguíneo desde el ventrículo izquierdo a la aorta rica en oxígeno.

## **10.2 AURICULAS**

Se podría definir las aurículas como las compuertas de ingreso del flujo sanguíneo al corazón.

Como tal son 2 aurículas, la izquierda, se encarga de captar la sangre llena de oxígeno desde los pulmones y la derecha recibe la sangre carente de oxígeno que regresa desde otras partes del cuerpo, para los procesos antes mencionados cada aurícula se descarga en su correspondiente ventrículo.

## **10.3 VENTRICULAS**

Los ventrículos son los cuartos inferiores del músculo cardíaco, se encarga de obtener la sangre que vienen desde las aurículas, por ejemplo, si se habla del ventrículo izquierdo este se encarga de recibir sangre enriquecida con oxígeno y transportarla hacia la aorta para ser establecida por todo el cuerpo, y si se habla de la aurícula derecha su función es entregar la sangre a los pulmones para oxigenarla.



FIGURA 4. FLUJO DE LA SANGRE PARTE 1. TOMADO DE EL APARATO CIRCULATORIO, POR CONRAD J. STORAD PAG 22

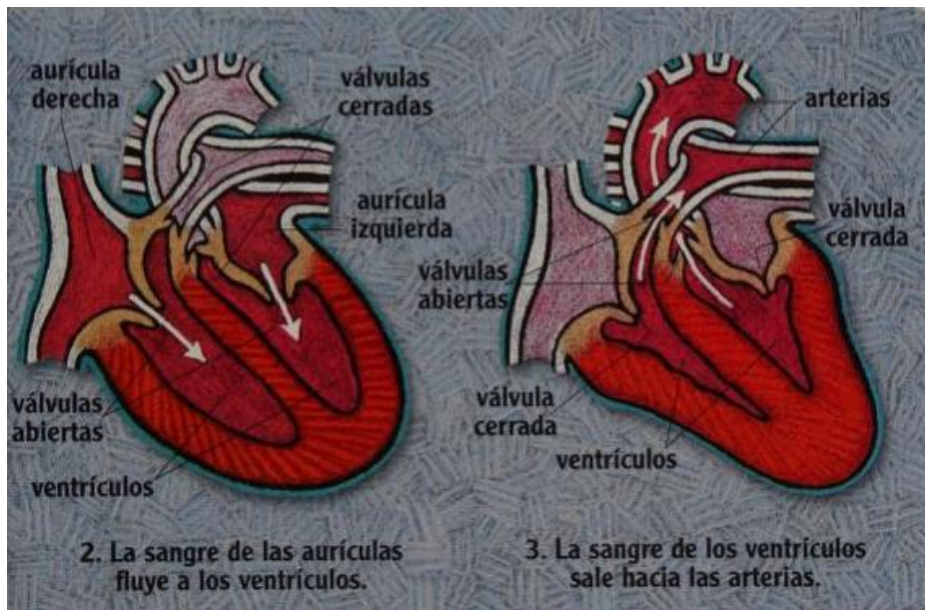


FIGURA 5. FLUJO SANGUÍNEO PARTE 2 Y 3 . TOMADO DE EL APARATO CIRCULATORIO, POR CONRAD J. STORAD PAG 23

Para ampliar más el entendimiento del corazón, es indispensable el análisis cauteloso y exacto, por eso se ha creado el electrocardiograma y el fono cardiograma.

#### 10.4 ELECTROCARDIOGRAMA

El electrocardiograma sencillamente es el conjunto de graficas que muestran el comportamiento del corazón, las señales ídicamente claramente como está trabajando,

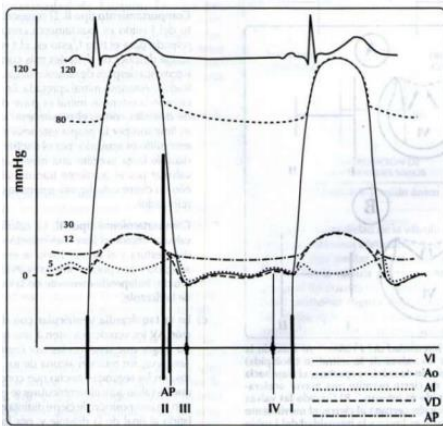


FIGURA 6 CICLO CARDIACO ,TOMADO DEL LIBRO CARDIOLOGÍA DE JOSE FERNANDO GUADALAJARA SEXTA EDICIÓN, PAG 43

Dentro del electrocardiograma se puede denotar lo siguiente:

Onda P. Se produce una despolarización auricular, con una duración de 0,10 segundo y un voltaje pico de 0,25 mV.

Ondas QRS. Se produce cuando se despolarizan los ventrículos del corazón, dura entre 0,6 a 10 milisegundos

Onda T. Se produce cuando se repolarizan los ventrículos.

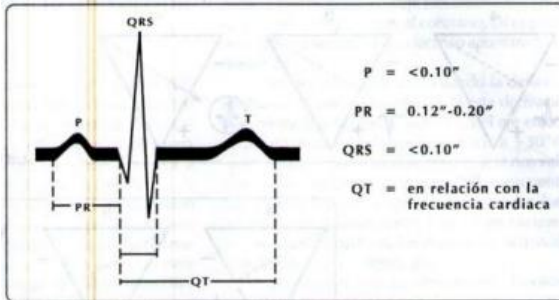


FIGURA 7 CONJUNTO ONDAS QUE CONFORMAN EL ELECTROCARDIOGRAMA , TOMADO DEL LIBRO CARDIOLOGÍA DE JOSE FERNANDO GUADALAJARA SEXTA EDICIÓN, PÁG. 75

## 10.5 FONOCARDIOGRAMA

La fonocardiografía es la adquisición de datos sonoros del corazón para mejorar la información obtenida en la auscultación cardíaca mediante un estetoscopio, además la información obtenida se podría procesar para derivar una información consistente que permite analizar los comportamientos del musculo principal circulatorio, esta data tan valiosa e interesante se denomina ruido.

Teniendo en cuenta las definiciones del corazón, sus comportamientos y formas de analizarlo se entrará más a fondo en la simulación de sus comportamientos.

## 11. SIMULACIÓN GRÁFICA DEL CORAZÓN

Teniendo en cuenta conceptos básicos acerca del corazón y de creación de software se crea un modelado gráfico del sistema cardiaco , eso que significa; Si se habla de sistema como un concepto abstracto y practico, se podría concebir como un conjunto de elementos a estudiar, con un conjunto de características y atributos y un conglomerado de comportamientos influenciables a través de ciertas entradas y salidas que indica como el sistema en si influye sobre su entorno , entonces se pensaría en el sistema cardiaco como un conjunto de entradas , en este caso la señales eléctricas , la sangre cargada de CO2 que entra a este y mediante un conjunto de comportamientos: la sístole y la diástole, generan otro acumulado de salidas , sangre oxigenada y purificada que se distribuye a todo el cuerpo humano.



Dentro del mundo científico es indispensable la experimentación sobre un sistema para entenderlo y manipularlo, sin embargo, se encuentran los principales problemas:

El experimento puede ser demasiado costoso, por ejemplo, se desea saber que les pasan a 100 carros aplastados por un avión, el experimento puede ser peligroso, ¿Qué pasaría si un ser humano se toma el agua lluvia acumulada por un caño?, ¿se experimentaría con la vida de un ser humano para adquirir dicha información?, dependiendo de ciertos valores morales y éticos determinadas personas tomarían su decisión, pero esto no deja de ser delicado.

Todos estos aspectos hacen que la simulación sea la vía ideal para la adquisición de información sobre un sistema, ya que se experimenta directamente sobre un modelo que no es más que un conjunto de estrategias y técnicas puestas a pruebas para la obtención de data relevante.

En la actualidad y más con la convivencia con entornos virtuales se hace necesario el entendimiento de la realidad sin necesidad de la exposición a una interacción directa, por ejemplo si se desease saber cuánta cantidad de agua puede haber en un edificio de 3 pisos, sería difícil la adquisición de una construcción para entregarla a ese tipo de escenarios, es por eso que es mucho más útil la simulación del comportamiento del agua sobre el edificio en estudio, ya que habría un conjunto de reglas y eventos que podrían replicarse y controlarse para su final comprensión.

La simulación del comportamiento del Corazón evidencia precisamente esta realidad, es difícil comprender su funcionamiento si no se posee el musculo humano, y mucho más si no se conocen distintas variables que pueden afectarlo.

Por ende, en este proyecto de grado, tiene como principal compromiso el estudio de las actividades cardiacas mediante sistemas electrónicos e informáticos, usando las siguientes herramientas:

Arduino, una excelente herramienta electrónica de código abierto que es usado en cientos de aplicaciones tecnológicas su IDE es multiplataforma lo que garantiza que pueda correr en distintos sistemas operativos, una de las principales características es que maneja librerías construidas en C++ y usa Microcontroladores AVR.

NodeJs, es un entorno de desarrollo en JavaScript usando V8 que es una herramienta de código abierto de google escrita en C++ que se usa en el navegador de Chrome.

JavaScript, es un lenguaje para desarrollo de software especificado primariamente para el desarrollo web, actualmente su popularidad ha aumentado ya que mediante distintos framework ha incursionado en otros contextos como el desarrollo backEnd, aplicativos de escritorio e incluso, interviene en el desarrollo electrónico.

Html5, es un lenguaje de maquetado que administra el contenido de las páginas web, con las nuevas especificaciones, tiene nuevos elementos multimedia.

Teniendo en cuenta estas definiciones, se adentra en lo que es el motivo de estudio que son los movimientos principales del corazón diástole, sístole y pulso, ya que un modelamiento total del corazón y detallar los aspectos puntuales del corazón no solo es difícil sino innecesario, por ende, se pasa a algo general y básico:

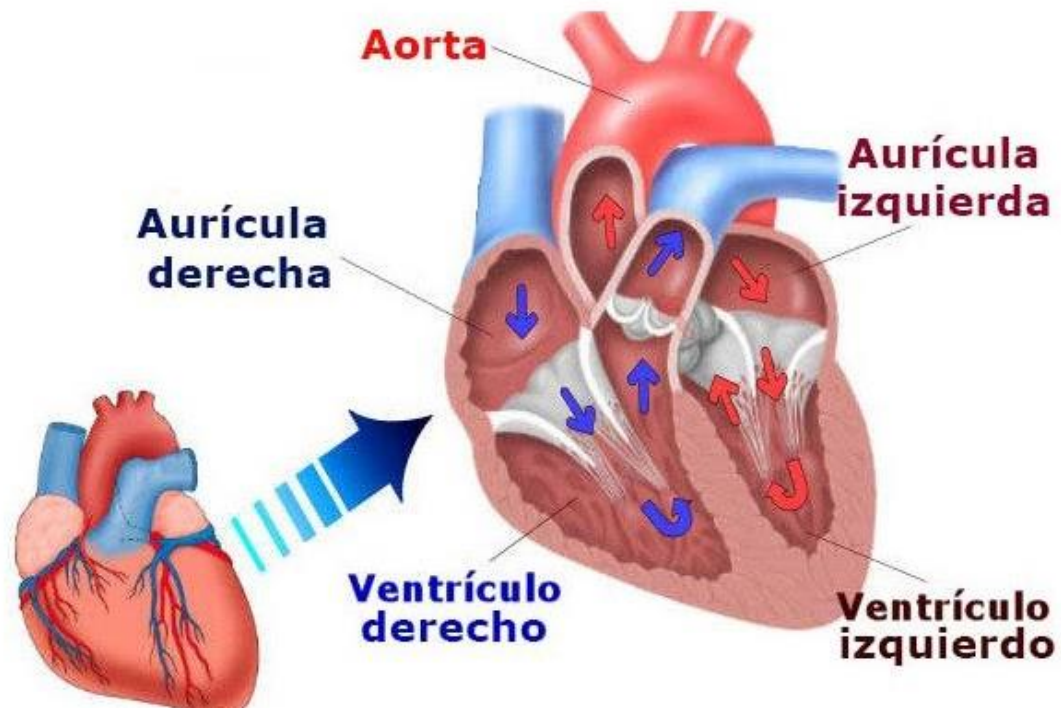
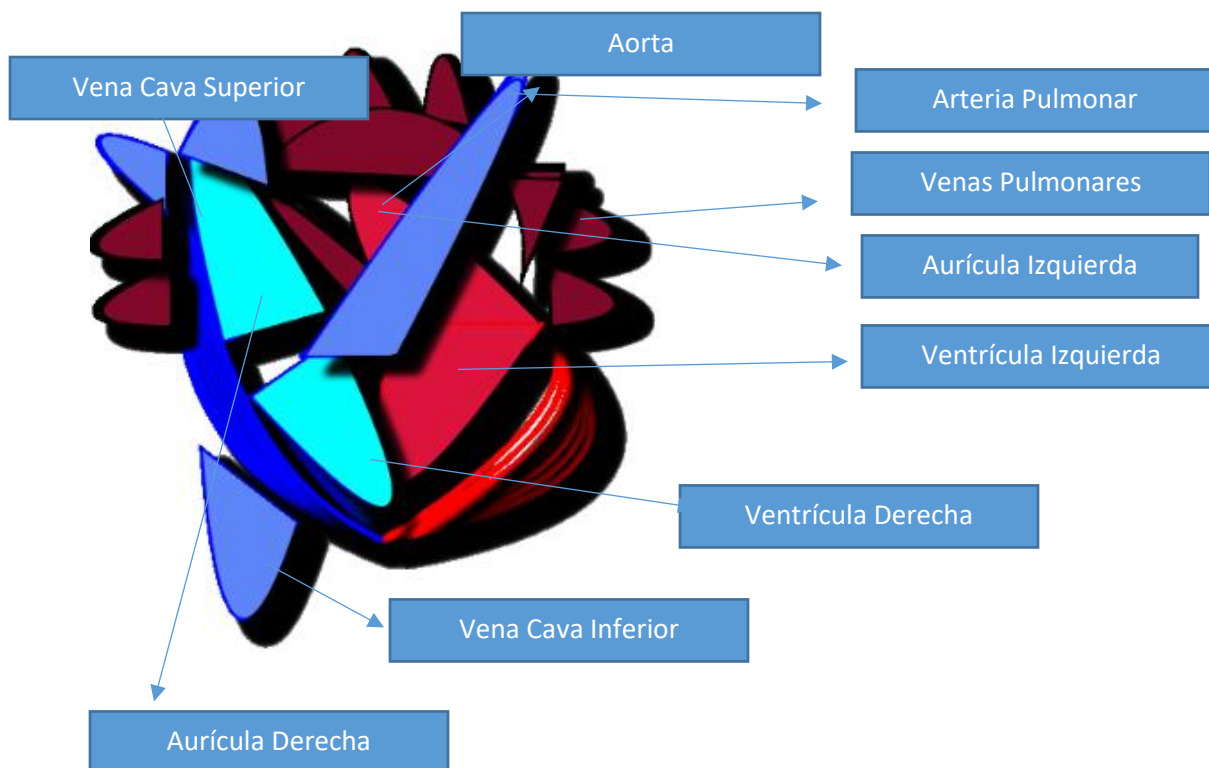
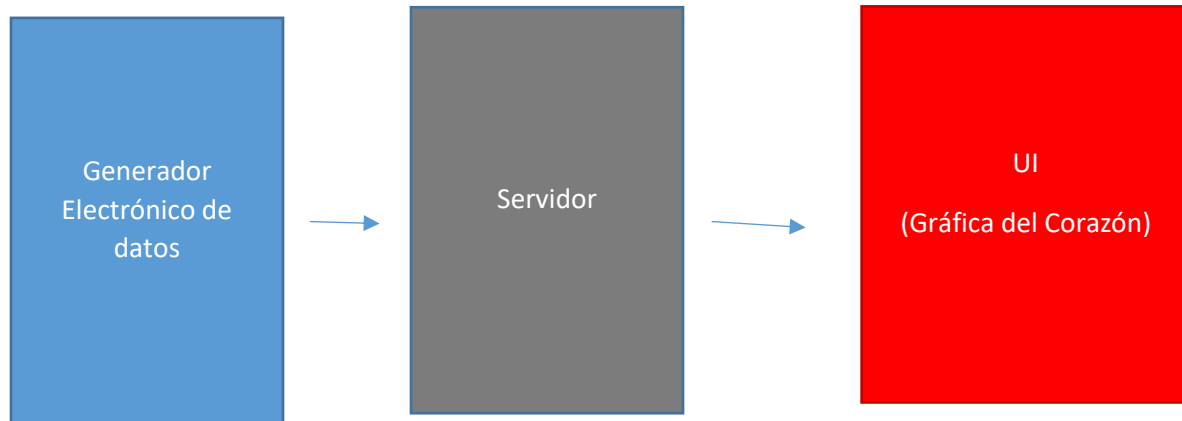


FIGURA 8, PARTES BÁSICAS DEL CORAZÓN. TOMADO DE LA WEB [HTTPS://PSICOLOGIAYMENTE.COM/SALUD/PARTES-DEL-CORAZON](https://psicologiaymente.com/salud/partes-del-corazon)

A una representación base de los elementos que permitirá analizar los movimientos antes mencionados:



De esa manera se contestará como se realiza la simulación en base a los siguientes bloques puntuales.



Generador Electrónico de datos:

Este bloque es un conjunto de datos enviados desde un microcontrolador AVR conectado a Arduino Uno, información que define el movimiento de los elementos gráficos del Corazón y la velocidad con la que estos se van a mover, simularía el pulso cardiaco y las distintas ondas que se detectan en un electrocardiograma, que irían directamente al servidor

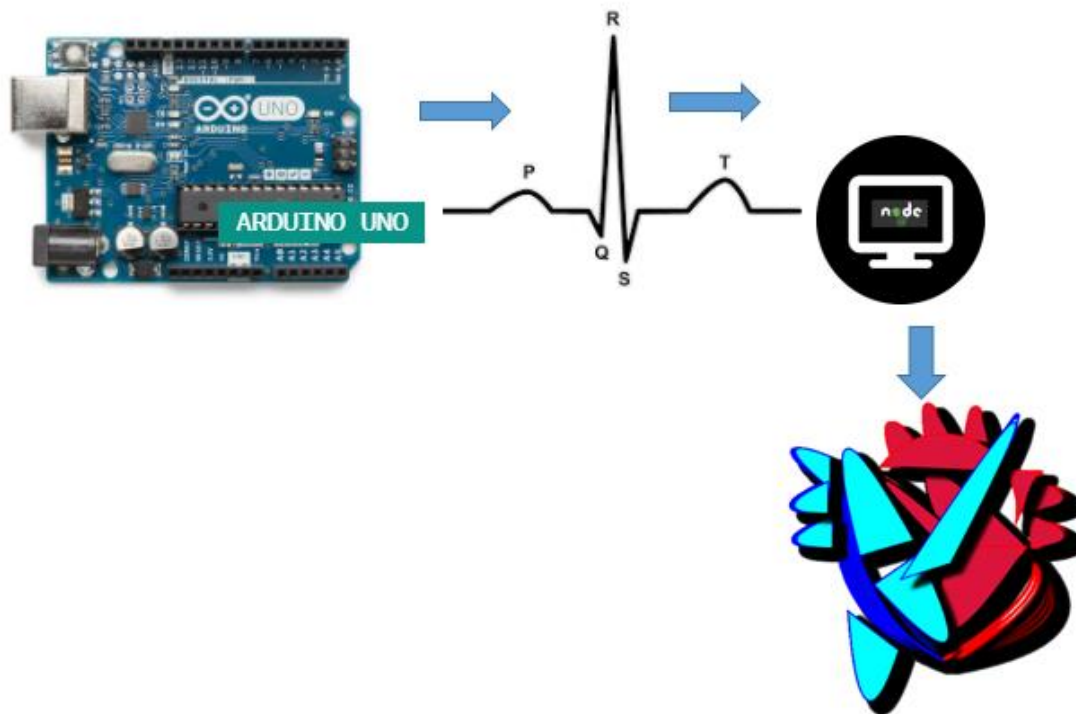


ILUSTRACIÓN 12. FLUJO GENERAL DE SIMULACIÓN DE MOVIMIENTOS CARDIACOS.FUENTE(AUTOR)

Servidor. Este bloque se encarga de comunicar la data enviada desde el arduino al bloque grafico que es un set de JavaScript embebido en HTML, usando NodeJs.

UI. Es la modelación grafica del corazón cuya naturaleza consiste en un conjunto de unidades graficas que simulan partes del musculo cardiaco mediante JavaScript y para ser visible en un entorno grafico es necesario poner dicho código dentro de HTML.

La creación de ese UI se basa en los siguientes bloques

Diagrama de Clases

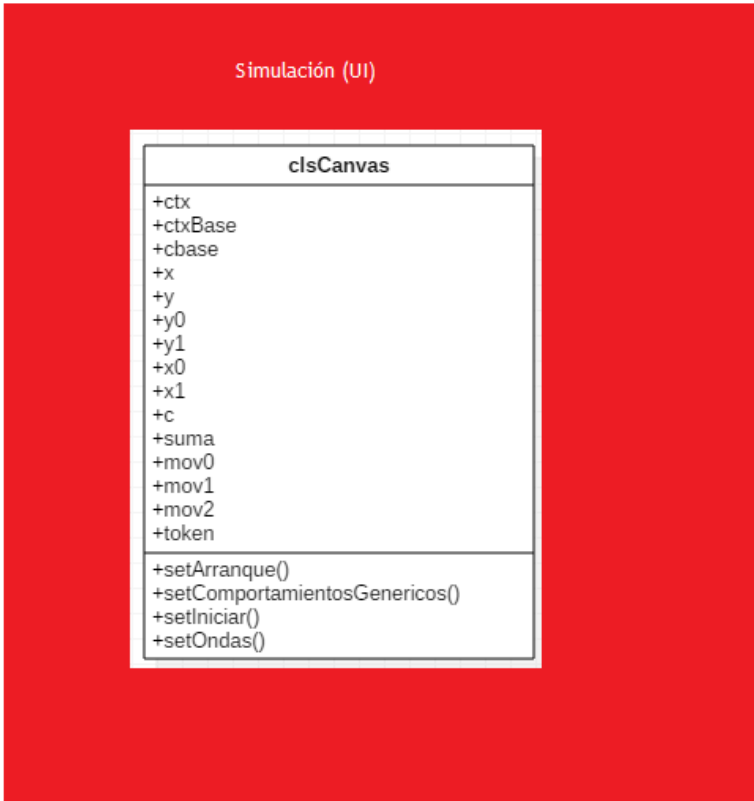


ILUSTRACIÓN 13. DIAGRAMA DE CLASES DE LA CLASE CLSCANVAS PARA LA SIMULACIÓN.FUENTE(AUTOR)

### Diagrama Básico Caso de Uso

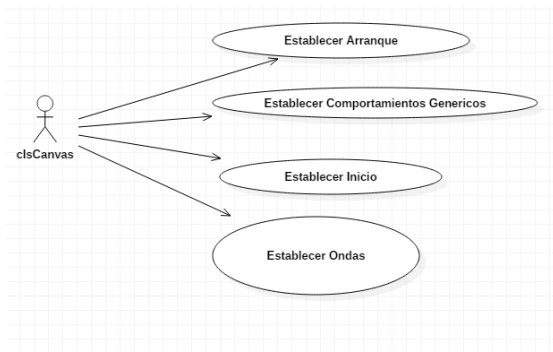


ILUSTRACIÓN 14. DIAGRAMA DE CASO DE USO PARA EL MANEJO DE LA CLASE CLSCANVAS.FUENTE(AUTOR)

## 12. CONCLUSIONES

Con el desarrollo de los diferentes procesos de modelamiento y simulación de comportamientos de ruidos cardiacos usando distintos elementos informáticos y electrónicos se puede lograr excelentes herramientas de estudio e investigación que pueden usar las diversas entidades y asociaciones científicas que se interesen por el mundo de la salud (Medicina, Biología, Bioingeniería).

Se evidencia el valor que tienen tecnologías relacionadas al software de alto rendimiento aplicado a las actividades que ejecuta la Fundación Jorge Reynolds donde se busca la explotación investigación de mecanismos y acciones que lleven a la sociedad el fruto de la ingeniería aplicada.

## 13. BIBLIOGRAFIA

Vélez A., Hernán, Rojas M, William, Borrero Ramírez, Jaime Restrepo Molina (2002) Fundamentos de Medicina Cardiología 6 ed. Editor Mario Montoya Toro.

Michael H. Crawford, Komandoor Srivathson, Dana P. McGlothlin (2007) Cardiología Consulta rápida. Editorial McGraw Hill Interamericana de España.

José Fernando Guadalajara (2006) Cardiología. Editorial Méndez Editores.

Ian Sommerville ,Victor Campos Olguin (2011) Ingeniería de software 9 ed. Editorial Pearson Educación.

Peter Fritzson (2005) Introducción al Modelado y Simulación de Sistemas Técnicos y Físicos con Modelica. Editorial Wiley-IEEE Press.

David A. Watt (1990) Programming Language Concepts and Paradigms. Editorial Prentice Hall.

Seyed H Roosta (2003) Foundations of Programming Languages. Design And Implementation .Editor Kallie Swanson.

NodeJs (2021) Inicio NodeJs. Obtenido de [www.nodejs.org](http://www.nodejs.org) :

<https://nodejs.org/es/>

Arduino (2021). What is Arduino?. Obtenido de [www.arduino.cc](http://www.arduino.cc):

<https://www.arduino.cc/en/Guide/Introduction>

JavaScript (2005). ¿Quieres convertirte en un desarrollador web front-end? Obtenido de [developer.mozilla.org](http://developer.mozilla.org):

<https://developer.mozilla.org/es/docs/Learn/JavaScript>

Sandro Fabián Castro Gutiérrez (15 de septiembre de 2021) videoSimulaciónCorazón

<https://www.youtube.com/watch?v=wycmMHq7TY>

Programación Lógica(2021) Historia de la programación Obtenido de [platzi.com](http://platzi.com)

<https://platzi.com/tutoriales/2211-historia-programacion/10285-programacion-logica/>

codigofacilito (5 de octubre de 2018) ¿Qué es la programación funcional?

[https://www.youtube.com/watch?v=kFQEL9Z24ZM&ab\\_channel=codigofacilito](https://www.youtube.com/watch?v=kFQEL9Z24ZM&ab_channel=codigofacilito)



## ANEXO CÓDIGO NODE.JS

```
io.on('connection', function(socket){
  console.log("apertura de conexion por socket");
});

const SerialPort = require('serialport');
var Delimiter = require('@serialport/parser-delimiter');

const mySerial = new SerialPort('COM3', {
  baudRate: 9600
});

const parser = new SerialPort.parsers.Readline();
mySerial.pipe(parser);
parser.on('data',(line) => {
  console.log(line)
  io.emit('arduino:data',{
    value: line.toString()
  });
});

mySerial.on('open', function () {
  console.log('puerto abierto');
});
```

```
app.get('/', (req, res) => {
  res.sendFile(__dirname + '/index.html');
});

server.listen(4000, () => {
  console.log("aquí arranca el servidor en puerto 4000");
});
```

## ANEXO CÓDIGO DE SIMULACIÓN GRÁFICA HTML - JAVASCRIPT

El código de la simulación es el siguiente.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Generador De Ruidos Cardiacos</title>
  </head>
  <body>
    <canvas id="myCanvas" width="500" height="550" style="border:1px solid #d3d3d3;">
      Si no se ve el canvas , cambiar el navegador.</canvas>

    <script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.4.0/Chart.min.js"></script>
    <script src="/socket.io/socket.io.js" charset="utf-8"></script>
    <script>
      const socket = io();
      socket.on('arduino:data',function(data){
```

```
console.log(data);
clsCanvas.setArranque(data);
});
var clsCanvas = {
  ctx : "",
  cbase : "",
  ctxBase : "",
  x : 0,
  y : 0,
  y0:0,
  y1:0,
  x0:0,
  x1:0,
  c:null,
  suma :0,
  mov0:[],
  mov1:[],
  mov2:[],
  token:false,
  setArranque : function(data){
    var arraybase = data.value;
    arraybase = arraybase.split(",");
    Objeto = Object.create(clsCanvas);
    clsCanvas.c = document.getElementById("myCanvas");
    clsCanvas.ctx = clsCanvas.c.getContext("2d");
    Objeto.ctx.beginPath();
    Objeto.c = document.getElementById("myCanvas");
    clsCanvas.ctx = Objeto.c.getContext("2d");
    clsCanvas.setComportamientosGenericas(5, "black", 15, 12 , 14, "");
```

```

        clsCanvas.setIniciar(arraybase);

    },

    setComportamientosGenericas : function(_tamannoSombra,
        _shadowColor, _shadowOffsetX, _shadowOffsetY,
        _lineWidth, _strokeStyle){

        clsCanvas.ctx.shadowBlur = _tamannoSombra;
        clsCanvas.ctx.shadowColor = _shadowColor;
        clsCanvas.ctx.shadowOffsetX = _shadowOffsetX;
        clsCanvas.ctx.shadowOffsetY = _shadowOffsetY;
        clsCanvas.ctx.lineWidth = _lineWidth;
        clsCanvas.ctx.strokeStyle = _strokeStyle;

    },

    setOndas : function(){

        clsCanvas.movPX = [0, 0, 0, 0, 0,0,0 ,0 ,0,0, 0,0 ,0,0,0,0 ,0,0,0 ,0 ];
        clsCanvas.movPY = [0, 0, 0, 0, 0,0,0 ,0 ,0,0, 0,0 ,0,0,0,0 ,0,0,0 ,0 ];
        clsCanvas.movP0 = [10,-10, 5 ,0, 0,0, 5 ,0, 0,0, -1,0 ,0,0,0,0 ,0,0,0 ,0 ];
        clsCanvas.movP1 = [10,-1 , 15 ,0, 0,0, 8 ,0, 0,4, 4,-2 ,0,0,0,0 ,0,0,0 ,2];
        clsCanvas.movP2 = [ 0,-1 ,-2 ,0, 0,0,-2 ,0, 0,0, -2,0 ,0,0,0,0 ,0,0,0 ,0];
        clsCanvas.movP3 = [ 0, 0 ,-0.5,0, 0,0,-0.2,0, 0,0, 2,0 ,0,0,0,0 ,0,0,0 ,0];
        clsCanvas.movsPQX = [0, 0, 0, 0, 0,0 ,0 ,0,0, 0,0 ,0,0,1,1,1 ,,-1,-1,0,0];
        clsCanvas.movsPQY = [0, 0, 0, 0, 0,0,0 ,0 ,0,0, 0,0 ,0,0,0,0,0 ,0,0,0 ,0];
        clsCanvas.movsPQ0 = [10,-10, 5 ,0, 0,0, 5 ,0, 0,0, -1,0 ,0,0,0,0 ,0,0,0 ,0];
        clsCanvas.movsPQ1 = [10,-1 , 15 ,0, 0,0, 8 ,0, 0,4, 4,-2,0,0,0,0 ,0,0,0 ,,-2];
        clsCanvas.movsPQ2 = [ 0,-1 ,-2 ,0, 0,0,-2 ,0, 0,0, -2,0 ,0,0,0,0 ,0,0,0 ,0];
    }

```

```

clsCanvas.movsPQ3 = [ 0, 0 ,-0.5,0, 0,0,-0.2,0, 0,0, 2,0 ,0,0,0,0 ,0,0,0 ,0];
clsCanvas.movQRSX = [0, 0, 0, 0, 0,0 ,0 ,0,0, 0,0 ,0,0,1,1,1 ,-1,-1,0,0];
clsCanvas.movQRSY = [0, 0, 0, 0, 0,0,0 ,0 ,0,0, 0,0 ,0,0,0,0 ,0,0,0,0 ];
clsCanvas.movQRS0 = [10,-10,5 ,0, 0,0,5 ,0, 0,0, -1,0 ,0,0,0,0 ,0,0,0 ,0];
clsCanvas.movQRS1 = [10,-1 ,15 ,0, 0,0,8 ,0, 0,4, 4,0 ,0,0,0,0 ,0,0,0 ,0];
clsCanvas.movQRS2 = [ 0,-1 ,-2 ,0, 0,0,-2 ,0, 0,0, -2,0 ,0,0,0,0 ,0,0,0 ,0];
clsCanvas.movQRS3 = [ 0, 0 ,-0.5,0, 0,0,-0.2,0, 0,0, 2,0 ,0,0,0,0 ,0,0,0 ,0];
clsCanvas.movsSTX = [0, 0, 0, 0, 0,0,0 ,0 ,0,0, 0,0 ,0,0,0,0 ,0,0,0 ,0];
clsCanvas.movsSTY = [0, 0, 0, 0, 0,0,0 ,0 ,0,0, 0,0 ,0,0,0,0 ,0,0,0 ,0];
clsCanvas.movsST0 = [0, 0, 0 ,0, 0,0,0 ,0, 0,0, 0,0 ,0,0,0,0 ,0,0,0 ,0];
clsCanvas.movsST1 = [0, 0, 0 ,0, 0,0,0 ,0.3, 0,0, 0,0 ,0,0,0,0 ,0,0,-0.1,0];
clsCanvas.movsST2 = [0, 0, 0 ,0, 0,0,0 ,1.5, 0,0, 0,0 ,0,0,0,0 ,0,0,-0.5 ,0];
clsCanvas.movsST3 = [0, 0, 0, 0, 0,0,0 ,0.2, 0,0, 0,0 ,0,0,0,0 ,0,0,-0.2,0];
clsCanvas.movTX = [0, 0, 0, 0.5,0.4,0.5,0 ,0,0.01 ,0,0, 0,0 , -0.2,0,0, 0 ,0,0,0 ,0];
clsCanvas.movTY = [0, 0, 0, 0, 0,0, 0 ,-1 ,0,0, 0,0 ,0,0,0,-1 ,0, 0, -0.01,0];
clsCanvas.movT0 = [0, 0, 0 ,0, 0,0, 0 ,0 ,0,0, 0,0 ,0,0,0,0 ,0,0,0 ,0];
clsCanvas.movT1 = [0, 0, 0 ,0, 0,0, 0 ,0, 0,0, 0,0 ,0,0,0,0 ,0,0,-2.1,0];
clsCanvas.movT2 = [0, 0, 0 ,0, 0,0, 0 ,0, 0,0, 0,0 ,0,0,0,0 ,0,0,1.5 ,0];
clsCanvas.movT3 = [0, 0, 0, 0, 0,0, 0 ,0.2, 0,0, 0,0 ,0,0,0,0 ,0,0,-0.2,0];
},

```

```

setIniciar : function(arraybase){

```

```

c = ["red","blue","aqua","crimson"," crimson","aqua","black","#5d85f7","#80092a"];
x = [350,100,200,200,250,300,351,300 ,250,150,180,89, 357,359,365,92, 86,250,100];
y = [300,200,290,155,155,160,300,320 ,300,350,438,220, 250,300,200,300,254,320,180];
x0 = [ 50,30 ,-140,-50, -50 ,-20,-200,-251, ,-50,158,-50,-89, 85,85,240,-92, -86,200,
50 ];

```

```

        y0 = [ 50 ,200,-250, -50, -100 ,-90,-200,-219, ,220,170,153,-75, -10,-10,250, 0, 0, -
380, -100];

        x1 = [-110,140, -70, -30, -30 ,-30,-100, -50, ,100, 58,-65, 0, 0, -1,100, 0, 0, -60 ,
60];

        y1 = [150 ,250, 20, 20, 20 , -5, 0, 0, ,0 ,-40,-50,-40, -40,-40,0 ,-40, -40, 2, 22];

        lineWidth = [4];

        strokeStyle = [c[0],c[1],c[1],c[6] ,c[6],c[6],c[6],c[6], c[1],c[0],c[1],c[1], c[1],c[6],c[6],c[6],
c[6],c[6],c[1],c[1]];

        fillStyle = [c[1],c[1],c[2],c[8] ,c[8],c[8],c[3],c[8], c[3],c[4],c[5],c[7], c[7],c[8],c[8],c[8],
c[8],c[8],c[7],c[7]];

        clsCanvas.setOndas();

        m = 0;

        for(var i = m; i < m+20; i++){

            arraybase[2] = arraybase[2].replace("\r","")

            clsCanvas.x = x[i] + clsCanvas["mov"+arraybase[2]+"X" ][i]*arraybase[1];

                clsCanvas.y = y[i] + clsCanvas["mov"+arraybase[2]+"Y" ][i]*arraybase[1];

            clsCanvas.ctx.lineWidth = lineWidth[i];

            clsCanvas.ctx.strokeStyle = strokeStyle[i];

            clsCanvas.ctx.fillStyle = fillStyle[i];

            clsCanvas.ctx.beginPath();

            clsCanvas.ctx.moveTo(clsCanvas.x, clsCanvas.y);

            if(i == 15){

                clsCanvas.ctx.arc(x0[i],y0[i],x1[i],1.2*Math.PI,2.03*Math.PI);

            }

            else{

                clsCanvas.ctx.quadraticCurveTo(clsCanvas.x + x0[i] + clsCanvas["mov"+arraybase[2]+"0"
][i]*arraybase[1], clsCanvas.y + y0[i] + clsCanvas["mov"+arraybase[2]+"1" ][i]*arraybase[1], clsCanvas.x
+ x1[i] + clsCanvas["mov"+arraybase[2]+"2" ][i]*arraybase[1],clsCanvas.y + y1[i] +
clsCanvas["mov"+arraybase[2]+"3" ][i]*arraybase[1]);

            }

            if(i > 1)clsCanvas.ctx.closePath();

```

```
        clsCanvas.ctx.stroke();  
        if(i > 1)clsCanvas.ctx.fill();  
    }  
}  
}  
  
</script>  
</body>  
</html>
```

## ANEXO CÓDIGO DE SIMULACIÓN GRÁFICA ARDUINO

```
char *cadena[] = {  
"0,1,sPQ",  
"0,1,sPQ",  
"0,1,sPQ",  
"1,1,P",  
"1,2,P",  
"2,3,P",  
"2,4,P",  
"1,5,P",  
"1,6,P",  
"0,6,sPQ",  
"0,6,sPQ",  
"0,6,sPQ",  
"0,6,sPQ",  
"-2,6,QRS",  
"-2,5,QRS",
```

```
"10,4,QRS",  
"20,3,QRS",  
"-6,2,QRS",  
"-3,1,QRS",  
"0,1,sST",  
"0,2,sST",  
"0,3,sST",  
"0,4,sST",  
"0,5,sST",  
"0,6,sST",  
"0,7,sST",  
"0,8,sST",  
"0,9,sST",  
"2,10,T",  
"2,11,T",  
"3,10,T",  
"4,9,T",  
"4,8,T",  
"3,7,T",  
"2,6,T"  
};
```

```
int carga = 0;
```

```
void setup() {  
  Serial.begin(9600);  
}
```

```
void loop() {
```



```
carga = carga + 1;  
if(carga == 35){  
    carga = 0;  
}  
Serial.println(cadena[carga]);  
delay(50);  
}
```