

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

**CARRERA:
INGENIERÍA DE SISTEMAS**

**Trabajo de titulación previo a la obtención del título de:
INGENIEROS DE SISTEMAS**

**TEMA:
REDISEÑO Y ELABORACIÓN ELECTRÓNICA PARA CONTROLAR Y
PROGRAMAR ESCENARIOS CLÍNICOS EN UN FANTOMA
OBSTÉTRICO DE
BAJA FIDELIDAD**

**AUTORES:
SERGIO ANDRÉS PALACIOS MONTENEGRO
ÁNGELA DAYANA ROMÁN CÁRDENAS**

**TUTOR:
JULIO RICARDO PROAÑO ORELLANA**

Quito, febrero de 2021

CESIÓN DE DERECHOS DE AUTOR

Nosotros Sergio Andrés Palacios Montenegro, con documento de identificación N° 1721326559 y Ángela Dayana Román Cárdenas, con documento de identificación N° 1725130890, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación intitulado: **REDISEÑO Y ELABORACIÓN ELECTRÓNICA PARA CONTROLAR Y PROGRAMAR ESCENARIOS CLÍNICOS EN UN FANTOMA OBSTÉTRICO DE BAJA FIDELIDAD**, mismo que ha sido desarrollado para optar por el título de: **INGENIEROS DE SISTEMAS**, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.



.....
SERGIO ANDRÉS PALACIOS MONTENEGRO

1721326559



.....
ÁNGELA DAYANA ROMÁN CÁRDENAS

1725130890

Quito, febrero de 2021

DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR

Yo declaro que bajo mi dirección y asesoría fue desarrollado el proyecto técnico, con el tema **REDISEÑO Y ELABORACIÓN ELECTRÓNICA PARA CONTROLAR Y PROGRAMAR ESCENARIOS CLÍNICOS EN UN FANTOMA OBSTÉTRICO DE BAJA FIDELIDAD** realizado por Sergio Andrés Palacios Montenegro y Ángela Dayana Román Cárdenas, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerado como trabajo final de titulación.

Quito, febrero del 2021



.....

JULIO RICARDO PROAÑO ORELLANA

CI: 0103909412

DEDICATORIA

Quiero dedicar esta tesis primeramente a Dios por haber permitido llegar hasta aquí hoy, por darme fuerza y salud para llevar a cabo mis metas y objetivos. Quiero darle las gracias por su amor infinito a mis padres, por haberme apoyado en cada uno de mis pasos y enseñarme buenos valores, por la motivación constante que permitieron que hoy en día sea la persona que soy y por su amor incondicional.

Sergio Andrés Palacios Montenegro

DEDICATORIA

A mis abuelitos Alicia y Galo, por ser mi guía, mi mayor apoyo y ser el tesoro más grande que tengo en la vida, por haber forjado la persona que ahora soy. A mis padres por permitir que cumpla una meta más, por nunca dejarme sola y por ser mi ejemplo para seguir. A mi familia por apoyarme en mi proceso de formación profesional y por su amor incondicional. A Franklin y mi segunda familia, por respaldarme y apoyarme, por el cariño que me demuestran día a día.

Ángela Dayana Román Cárdenas

AGRADECIMIENTO

Nuestros profundos agradecimientos a la Universidad Politécnica Salesiana que fue el ente principal en nuestra formación como profesional, a nuestros Ingenieros por todos los conocimientos impartidos que ayudaron a formar profesionales con ética y valores. A nuestros amigos que formaron parte de nuestro proceso académico y los momentos compartidos logrando alcanzar una meta en común, a nuestro tutor el Ingeniero Julio Ricardo Proaño Orellana por su apoyo incondicional y por el seguimiento en cada etapa del proyecto.

Sergio Andrés Palacios Montenegro

Ángela Dayana Román Cárdenas

ÍNDICE DE CONTENIDOS

CAPÍTULO I	1
INTRODUCCIÓN	1
1.1 Antecedentes	1
1.2 Planteamiento del Problema	1
1.3 Justificación	2
1.4 Objetivos	3
1.5 Objetivo general.....	3
1.6 Objetivos específicos	3
1.7 Metodología	3
1.8 Planificación	4
1.9 Diseño	5
1.10 Desarrollo	5
1.11 Pruebas.....	5
CAPÍTULO II	6
MARCO TEÓRICO	6
2.1 Sistemas Embebidos	6
2.1.1 Clasificación de los Sistemas Embebidos	7
2.1.1.1 Sistemas embebidos independientes.....	7
2.1.1.2 Sistemas embebidos en tiempo real.....	7
2.1.2 Arquitectura	8
2.1.2.1 Componentes software.....	8
2.1.2.2 Componentes hardware	8
2.1.3 Lenguajes de programación	9
2.1.4 Lenguaje ANSI C.....	10
2.1.5 Aplicaciones	11
2.2 Arduino	12
2.2.1 Placas Arduino	13
2.2.2 Comandos AT	14
2.3 Tecnologías Web	16
2.3.1 Páginas Web Estáticas	17
2.3.2 Páginas Web Dinámicas.....	18
2.3.3 Arquitectura básica de una aplicación Web	19
.....	21
2.3.4 El Modelo Cliente / Servidor	21
2.3.5 Cliente	22
2.3.6 Servidor	23
2.4 Lenguajes de Programación Web	23
2.4.1 Lenguaje HTML	24

2.4.2	Lenguaje CSS.....	24
2.4.3	Lenguaje JavaScript	25
2.5	Fantoma Obstétrico de Baja Fidelidad	26
2.5.1	Modelo de palpación abdominal	27
	27
2.6	Simulación de emergencias obstétricas	28
2.6.1	Escenarios del uso de un Fantoma Obstétrico de Baja Fidelidad	28
2.6.1.1	Hemorragia postparto	29
2.6.1.2	Preeclampsia.....	29
2.6.1.3	Eclampsia.....	29
2.6.1.4	Maniobras de Leopold.....	29
CAPÍTULO III.....		30
ALCANCE DEL PROYECTO		30
3.1	Análisis de factibilidad	30
3.1.1	Factibilidad Técnica	30
3.1.1.1	Hardware.....	31
3.1.1.2	Software.....	32
3.1.2	Factibilidad Operativa.....	33
3.1.2.1	Recursos Humanos	33
3.2	Resultados esperados	34
CAPÍTULO IV		36
PLANIFICACIÓN		36
4.1	Análisis de requerimientos	36
4.2	Historias de Usuarios	38
4.3	Asignación de Roles de Usuario.....	43
4.4	Plan de entrega del Proyecto.....	43
4.5	Iteraciones	44
4.5.1	Primera Iteración.....	44
4.5.1.1	Tareas de Ingeniería.....	44
4.5.2	Segunda Iteración.....	47
4.5.2.1	Tareas de Ingeniería.....	47
4.5.3	Tercera Iteración	51
4.5.3.1	Tareas de Ingeniería.....	51
CAPÍTULO V.....		57
DISEÑO		57
5.1	Diagrama de bloques del proceso de comunicación del simulador	57
5.2	Diagramas UML	58
5.2.1	Diagramas de casos de uso.....	59

5.2.2	Diagrama de estado	63
5.3	Tarjeta CRC (Clase – Responsabilidades - Colaboradores)	65
5.4	Bosquejo del Diseño	66
5.4.1	Bosquejo de la Página Web	66
CAPÍTULO VI.....		67
DESARROLLO.....		67
6.1	Análisis de estado inicial del Fantoma.	67
6.1.1	Análisis de la cabeza del simulador	68
6.1.1.1	Ojos.....	68
6.1.1.2	Boca.....	68
6.1.2	Análisis de la cavidad torácica del simulador.	69
6.1.3	Análisis de los brazos del simulador.	70
6.1.4	Análisis de la cavidad abdominal del simulador.	70
6.2	Análisis de las tarjetas Arduino	72
6.3	Configuración del módulo de comunicación	74
6.4	Interfaz Desarrollada.....	75
6.4.1	Herramientas del controlador.	76
6.4.2	Contenedores de control para activación del simulador	77
6.4.3	Contenedor de sonidos torácicos.....	77
6.4.4	Contenedor de sonidos de la paciente	78
CAPÍTULO VII.....		79
RESULTADOS.....		79
7.1	Pruebas de Aceptación.....	79
7.1.1	Pruebas de la primera iteración	79
7.1.1.1	Descripción de pruebas.....	80
7.1.2	Pruebas de la segunda iteración	82
7.1.2.1	Descripción de pruebas.....	82
7.1.3	Pruebas de la tercera iteración.....	85
7.1.3.1	Descripción de pruebas.....	85
7.2	Validación del Proyecto.....	89
CONCLUSIONES.....		90
RECOMENDACIONES.....		91
REFERENCIAS		92
ANEXOS		97

ÍNDICE DE TABLAS

Tabla 1 Campos y aplicaciones que han utilizado sistemas embebidos	11
Tabla 2 Placas Arduino	13
Tabla 3 Comandos AT	15
Tabla 4 Especificaciones mínimas de una PC.....	31
Tabla 5 Especificaciones del módulo Bluetooth	31
Tabla 6 Especificaciones del módulo Wi-Fi	32
Tabla 7 Recursos Humanos.....	33
Tabla 8 Requerimientos Funcionales.	36
Tabla 9 Requerimientos no funcionales.....	38
Tabla 10 Historia de Usuario Acceso al Sistema	39
Tabla 11 Historia de Usuario Gestión Multimedia	39
Tabla 12 Historia de Usuario Gestión de movimientos de ojos.....	40
Tabla 13 Historia de Usuario Gestión de Sonidos Torácicos	40
Tabla 14 Historia de Usuario Gestión de Pulsos Cardíacos.....	41
Tabla 15 Historia de Usuario Gestión de Labios	42
Tabla 16 Historia de Usuario Gestión de Bomba de Sangre.....	42
Tabla 17 Asignación de roles de usuario	43
Tabla 18 Plan de entrega del Proyecto	43
Tabla 19 Primera Iteración.....	44
Tabla 20 Tareas de Ingeniería	44
Tabla 21 Tarea de Ingeniería 1 para la Iteración 1	45
Tabla 22 Tarea de Ingeniería 2 para la Iteración 1	45
Tabla 23 Tarea de Ingeniería 3 para la Iteración 1	46
Tabla 24 Tarea de Ingeniería 4 para la Iteración 1	46
Tabla 25 Tarea de Ingeniería 5 para la Iteración 1	47
Tabla 26 Segunda Iteración.....	47
Tabla 27 Tareas de Ingeniería	48
Tabla 28 Tarea de Ingeniería 1 para la Iteración 2.....	48
Tabla 29 Tarea de Ingeniería 2 para la Iteración 1	49
Tabla 30 Tarea de Ingeniería 3 para la Iteración 1	49
Tabla 31 Tarea de Ingeniería 4 para la Iteración 2.....	50
Tabla 32 Tarea de Ingeniería 5 para la Iteración 2.....	50
Tabla 33 Tarea de Ingeniería 6 para la Iteración 2.....	51
Tabla 34 Tercera Iteración	51
Tabla 35 Tareas de Ingeniería	52
Tabla 36 Tarea de Ingeniería 1 para la Iteración 3.....	52
Tabla 37 Tarea de Ingeniería 2 para la Iteración 3.....	53
Tabla 38 Tarea de Ingeniería 3 para la Iteración 3.....	53
Tabla 39 Tarea de Ingeniería 4 para la Iteración 3.....	54
Tabla 40 Tarea de Ingeniería 5 para la Iteración 3.....	54
Tabla 41 Tarea de Ingeniería 6 para la Iteración 3.....	55
Tabla 42 Tarea de Ingeniería 7 para la Iteración 3.....	55
Tabla 43 Tarea de Ingeniería 8 para la Iteración 3.....	56

Tabla 44 Tarea de Ingeniería 9 para la Iteración 3.....	56
Tabla 45 Tarjeta CRC Control Fantoma	65
Tabla 46 Parámetros del simulador.....	73
Tabla 47 Caso de pruebas primera iteración	79
Tabla 48 Caso de prueba acceso al sistema	80
Tabla 49 Caso de prueba gestión de credenciales multimedia.....	81
Tabla 50 Caso de pruebas segunda iteración	82
Tabla 51 Caso de prueba gestión de movimiento de ojos	83
Tabla 52 Caso de prueba gestión de labios	84
Tabla 53 Caso de pruebas tercera iteración.....	85
Tabla 54 Caso de prueba gestión de pulsos cardíacos	85
Tabla 55 Caso de prueba gestión de movimientos torácicos	87
Tabla 56 Caso de prueba gestión de bomba de sangre.....	88

ÍNDICE DE FIGURAS

Figura 1 Ciclo de la Metodología XP	4
Figura 2 Arquitectura formada de SW y HW de un sistema embebido heterogéneo Fuente: (Camargo Bareño, 2011).....	9
Figura 3 Componentes de la placa Arduino	13
Figura 4 Diferencias entre Web 1.0 y Web 2.0 Fuente (Rodríguez-Palchevich, 2008)	17
Figura 5 Funcionamiento de una página Web estática.....	18
Figura 6 Funcionamiento de una Arquitectura Web	21
Figura 7 Flujo del Modelo Cliente/Servidor	22
Figura 8 Prototipo Obstétrico diseñado en el siglo XVIII	27
Figura 9 Modelo abdominal de palpación.....	27
Figura 10 Ejemplo de una técnica híbrida que combina un simulador obstétrico de baja fidelidad con un actor.	28
Figura 11 Esta figura muestra el proceso de comunicación que tiene el simulador y la página Web	58
Figura 12 . Esta figura muestra el caso de uso general de la página Web con sus especificaciones generales	59
Figura 13 Esta figura muestra el caso de uso específico referente al movimiento de ojos	60
Figura 14 Esta figura muestra el caso de uso específico referente a los sonidos del tórax	60
Figura 15 Esta figura muestra el caso de uso específico referente a los pulsos cardíacos.....	61
Figura 16 Esta figura muestra el caso de uso específico referente a los estados de labios.	61
Figura 17 Esta figura muestra el caso de uso específico referente a los estados del sangrado.	62
Figura 18 Esta figura muestra el caso de uso específico referente a los sonidos de la paciente.	62
Figura 19 Esta figura muestra el diagrama de estado que tiene la página Web.....	63
Figura 20 Esta figura indica el ciclo inicial de ingreso y validación de IP.	64
Figura 21 Esta figura indica el ciclo al momento que el usuario envía una petición. 64	
Figura 22 Esta figura muestra el bosquejo principal del diseño de la página Web. ..	66
Figura 23 Esta figura muestra una vista panorámica de los componentes a analizar. 67	
Figura 24 Esta figura muestra los elementos que permiten el funcionamiento de los ojos del simulador.	68
Figura 25 Esta figura muestra los elementos que permiten el funcionamiento de los ojos del simulador.	69
Figura 26 Esta figura muestra la ubicación de los parlantes tanto del corazón como de los pulmones.....	69
Figura 27 Esta figura muestra la ubicación de los motores vibradores en cada uno de los brazos.....	70
Figura 28 Esta figura muestra la distribución de todas las placas en la cavidad abdominal.....	71

Figura 29 Esta figura muestra todos los parámetros enviados a cada uno de los actuadores del simulador.	72
Figura 30 Esta figura muestra el resultado y configuración del módulo de comunicación ESP8266.	75
Figura 31 Esta figura muestra la interfaz Web donde el usuario podrá controlar al simulador.	76
Figura 32 Esta figura muestra las herramientas que podrá usar el usuario para gestionar los sonidos de la paciente.	76
Figura 33 Esta figura indica todos los controles que el usuario podrá realizar sobre los diferentes actuadores del simulador.	77
Figura 34 Esta figura indica el contenedor de sonidos torácicos que el usuario puede enviar al simulador.	77
Figura 35 Esta figura indica el contenedor de sonidos que la paciente podrá emitir dependiendo de la complicación médica.	78
Figura 36 Esta figura muestra la correcta validación del caso de prueba acceso al sistema por parte del ingeniero a cargo de la clínica de simulación.	81
Figura 37 Esta figura muestra la correcta validación del caso de prueba gestión de movimiento de ojos por parte del ingeniero a cargo de la clínica de simulación.	83
Figura 38 Esta figura muestra la correcta validación del caso de prueba gestión de movimiento de labios por parte del ingeniero a cargo de la clínica de simulación. ..	84
Figura 39 Esta figura muestra la correcta validación del caso de prueba gestión de pulsos cardíacos por parte del ingeniero a cargo de la clínica de simulación.	86
Figura 40 Esta figura muestra la correcta validación del caso de prueba gestión de movimientos torácicos por parte del ingeniero a cargo de la clínica de simulación. .	87
Figura 41 Esta figura muestra la correcta validación del caso de prueba gestión de bomba de sangre por parte del ingeniero a cargo de la clínica de simulación.	88

RESUMEN.

Con el pasar de los años el uso de simuladores médicos en la formación profesional de los estudiantes ha ido evolucionando como es el caso de los fantomas obstétricos, con el objeto de simular las complicaciones médicas que se puedan presentar en el proceso del parto.

El presente proyecto de titulación específica el rediseño e implementación de un prototipo eléctrico basado en una comunicación Wi-Fi entre el maniquí obstétrico y la página Web, logrando mejorar los procesos de simulación desarrollados en la clínica de simulación de la Universidad Central del Ecuador y a su vez potenciar el desarrollo de las habilidades de estudiantes obtenidas de dichas prácticas.

En el capítulo 1 se plantea la problemática, los objetivos y se describe la metodología XP la cual se tomó como referencia para la ejecución del proyecto, en el capítulo 2 se analiza los conceptos necesarios que proporcionan al lector entendimiento sobre el desarrollo del proyecto. En el tercer capítulo se analiza el alcance del proyecto y se realiza un análisis de factibilidad para identificar la viabilidad del mismo.

En los capítulos 4,5,6 y 7 se desarrolla el proyecto en basa a la metodología implementada, la etapa de planificación cuenta con un análisis de requerimientos, en la etapa de diseño se realiza un bosquejo de las interfaces y los respectivos diagramas uml, en la etapa de desarrollo se realiza un análisis del estado inicial del fantoma, finalmente en la fase de resultados se lleva a cabo las pruebas de aceptación del proyecto.

ABSTRACT

Over the years, the use of medical simulators has evolved in students' professional training, as is the case of obstetric simulators, to simulate the medical complications that can occur in the birth process. This degree project specifies the redesign and implementation of an electrical prototype based on Wi-Fi communication between the obstetric mannequin and the Website, improving the simulation processes developed in The Central University of Ecuador's simulation clinic development of student skills obtained from these practices.

Chapter 1 describes the problem, the objectives, and the XP methodology, taken as a reference for the project's execution. Chapter 2 analyzes the necessary concepts that provide the reader with an understanding of the project's development. In the third chapter, the project's scope is analyzed, and feasibility analysis is carried out to identify its viability. In chapters 4, 5, 6, and 7, the project is developed based on the implemented methodology. The planning stage includes a requirements analysis. In the design stage, a sketch of the interfaces and the selected UML diagrams is made. In the development stage, an analysis of the initial state of the simulator. Finally, in the results phase, the acceptance tests of the project are carried out.

CAPÍTULO I

INTRODUCCIÓN

1.1 Antecedentes

En la universidad central del Ecuador se encuentra la clínica de simulación y robótica donde se ubican tanto laboratorios como simuladores clínicos, permitiendo mejorar todas las practicas medicas de la comunidad universitaria, esto ha logrado que todos los estudiantes de medicina accedan a los diferentes servicios que ofrece la clínica, como consecuencia se ve la necesidad de mejorar el funcionamiento y acceso a los simuladores.

Varias universidades han logrado implementar el uso de simuladores médicos los cuales se han revisado para determinar sus falencias tanto electrónicas como de acceso de esta forma se logrará tomar las mejores decisiones y poder rediseñarlas. Ejemplo de esto es la Universidad Católica de Guayaquil (Carriel & Zambrano, 2014) y la Universidad Central del Ecuador (Falcón Ayala, 2016).

1.2 Planteamiento del Problema

El avance de la tecnología ha permitido incorporar nuevas herramientas que permiten desarrollar destrezas en el aprendizaje, como es el caso de los fantasmas. Estos elementos ayudan a simular casos médicos que suceden en pacientes, para emular complicaciones médicas en diferentes escenarios. Los estudiantes usan un software de escritorio el cual permite simular las complicaciones postparto como: Preeclampsia, Eclampsia, Maniobras de Leopold y Hemorragia postparto, de esta manera pueden desarrollar sus habilidades en este tipo de entornos y no en pacientes reales.

Actualmente la Clínica de Simulación Médica y Robótica de la Facultad de Ciencias Médicas de la Universidad Central del Ecuador cuenta con escenarios clínicos para la formación práctica de sus estudiantes, estos escenarios poseen fantomas robots. Cada fantoma tiene un software el cual se conecta mediante Bluetooth para la comunicación y envío de datos, el problema es la intermitencia en la conexión que impide tener una correcta simulación de un escenario. Este problema es debido a la tasa de velocidad de transferencia del módulo Bluetooth que se está utilizando, lo que produce una transferencia de datos lenta.

Este proyecto tiene como objetivo implementar un módulo electrónico Wi-Fi y el diseñar un sistema informático, los cual permitirá mejorar la comunicación y el control del simulador, logrando de esta manera eliminar la instalación de aplicaciones en los ordenadores para controlar el simulador, mejorando el QoE (Calidad de experiencia) y logrando un tiempo de respuesta adecuado.

1.3 Justificación

La implementación de un módulo de comunicación Wi-Fi que sustituirá al módulo bluetooth actual logrará mejorar la transferencia de información, y el desarrollo de un sistema informático permitirá que exista una mejor gestión al trabajar con el simulador. Los estudiantes e instructores podrán realizar sus prácticas de manera efectiva, de este modo se estará reduciendo en un gran porcentaje las fallas de conexión y a su vez proporcionará una respuesta inmediata al interactuar de forma directa entre la página Web y el simulador.

La mejora en la comunicación conseguirá que el tiempo de respuesta entre el practicante y el simulador sea efectivo, y pueda tomar las acciones necesarias dependiendo el escenario en el que se encuentre, y logrando una experiencia que se asemeje a la realidad.

1.4 Objetivos

1.5 Objetivo general

Rediseñar y elaborar un modelo eléctrico Wi-Fi y un sistema informático para programar escenarios clínicos en un fantoma obstétrico de baja fidelidad.

1.6 Objetivos específicos

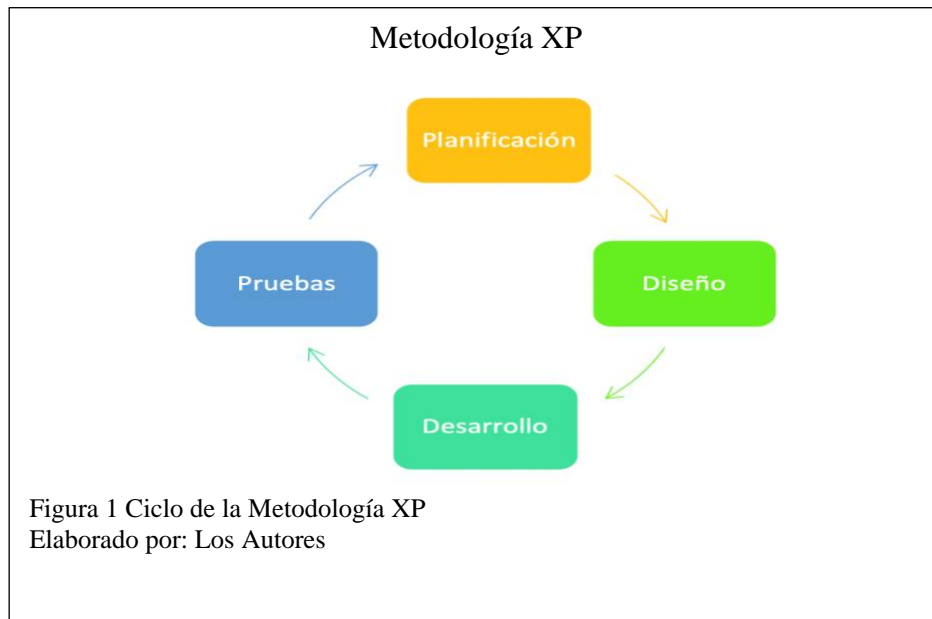
- Construir una plataforma Web que permita simular escenarios clínicos en un fantoma obstétrico de baja fidelidad en la Clínica de Simulación Médica y Robótica de la Universidad Central del Ecuador.
- Implementar un módulo Wi-Fi que ayude a mejorar la comunicación y conexión entre el fantoma y el sistema informático.
- Proporcionar un software confiable, eficiente e intuitivo que permita mejorar el (QoE) del usuario.

1.7 Metodología

Para llevar a cabo el desarrollo del proyecto se debe elegir una metodología que se acople a las especificaciones y características de este, es por eso que se ha elegido una metodología ágil denominada extreme programming (XP). Esta metodología permite realizar cambios, tener una programación organizada y realizar pruebas conforme avanza el proyecto. A continuación, se definirá dicha metodología.

El objetivo principal de esta metodología es la retroalimentación continua entre el usuario y el equipo de desarrollo, de esta manera se logra enfocar los requerimientos por parte del cliente y plantearlos de una forma precisa en el desarrollo de la página

Web. Se fundamenta en un flujo de procesos, los cuales se ira detallando a continuación.



1.8 Planificación

En esta etapa se planificará una reunión con el encargado de la Clínica de Simulación y los desarrolladores a cargo del proyecto, donde se establecerán todas las funciones que la página y el simulador deberán tener.

En esta sección se tendrá en cuenta 5 elementos importantes que son:

- Análisis de Requerimientos.
- Historias de Usuarios.
- Asignación de Roles de Usuario.
- Plan de entrega del Proyecto.
- Iteraciones.

Los cuales se detallarán a profundidad en el Capítulo 5.

1.9 Diseño

En esta etapa se planificará una guía de diseño, la misma que debe ser clara y simple de entender tanto por el desarrollador como por el docente (usuario), de esta forma se obtendrá el prototipo que será revisado por el cliente y el cual estará sujeto a cambios. Esto se detallará de mejor manera en el Capítulo 6

1.10 Desarrollo

En esta etapa se implementará la codificación y el desarrollo en parejas donde se manejará una programación estructurada, en la que cada uno de los integrantes del proyecto podrán realizar una tarea en específico, logrando abarcar todos los requerimientos solicitados por el cliente, de igual manera se adjuntara las evidencias del código y de la página más relevantes. Este proceso se explicará a detalle en el Capítulo 7.

1.11 Pruebas

En esta última etapa la cual se desarrollará en el Capítulo 8 se establecerán todas las medidas de comprobación referentes a la funcionalidad de cada uno de los componentes que se desarrollaron, estas pruebas serán realizadas por el ingeniero a cargo de la clínica de simulación.

Esto es primordial ya que es necesario corregir todos los errores que se puedan encontrar para entregar un software confiable.

CAPÍTULO II

MARCO TEÓRICO

Este capítulo busca proporcionar al lector los conceptos básicos y necesarios para la comprensión del desarrollo de este proyecto.

En primera instancia, se explica la definición de sistemas embebidos con la finalidad de comprender cómo estos sistemas han permitido optimizar recursos y enfocarse en tareas específicas de manera eficaz. La optimización de estos recursos permitirá llevar a cabo el proceso de la presente investigación.

Posteriormente, se describe todo lo referente a Arduino, se conocerán los tipos de placas que se manejan actualmente, así como también los comandos que permiten que su configuración sea sencilla; esto se realiza con el fin de conocer sus diferentes usos y saber de qué manera se pueden implementar.

De igual manera, se describen las tecnologías Web más utilizadas y los tipos de tecnologías empleadas, específicamente la Web dinámica y estática que serán el objeto de este estudio.

Por último, se realizará un recuento de los fantomas obstétricos que serán utilizados para llevar a cabo el desarrollo del proyecto.

2.1 Sistemas Embebidos

Para (Arriarán, 2017) un sistema embebido es un circuito digital de propósito específico capaz de realizar operaciones de computación mediante un conjunto de instrucciones que ayuden a cumplir una determinada tarea. A diferencia de un

computador de propósito general, los sistemas embebidos están diseñados para cumplir una función o un rango de funciones y no para ser programados por el usuario final. Si bien el usuario podrá tomar decisiones sobre su funcionalidad, pero no podrá cambiarlas. (Heath, 2002)

Según el autor (Camargo Bareño, 2011) un sistema embebido está formado por componentes hardware como circuito integrado de aplicaciones específicas (ASIC) y dispositivo lógico programable (PLD), así también software como microprocesadores, microcontroladores y procesador digital de señales (DSP). Existen algunos tipos de sistemas embebidos que serán descritos en el siguiente punto.

2.1.1 Clasificación de los Sistemas Embebidos

(Hernández Camacho, 2016) menciona que los sistemas embebidos se clasifican en dos grandes grupos; sistemas embebidos en tiempo real y sistemas embebidos independientes.

2.1.1.1 Sistemas embebidos independientes

Estos sistemas funcionan en modo independiente, es decir toman entradas que pueden ser señales eléctricas las cuales son procesadas, obteniendo un resultado esperado. Generalmente son usados en artículos electrónicos de consumo, automóviles o para controlar procesos.

2.1.1.2 Sistemas embebidos en tiempo real

Su característica principal es que cuenta con un período de tiempo específico y limitado para poder cumplir con su función, esto les permite reaccionar ante cualquier problema presentado debido a que se asigna un plazo de tiempo definitivo. A diferencia de los sistemas independientes, estos son utilizados en aplicaciones sofisticadas y que necesitan que la captura de sus señales y sus acciones de control

sean en tiempo real debido a su complejidad (Hernández Camacho, 2016). A continuación, se describe la arquitectura de estos sistemas.

2.1.2 Arquitectura

Para los autores (Camargo et al., 2012), un sistema embebido está compuesto tanto por software como hardware como se muestra en la Figura 2.

2.1.2.1 Componentes software

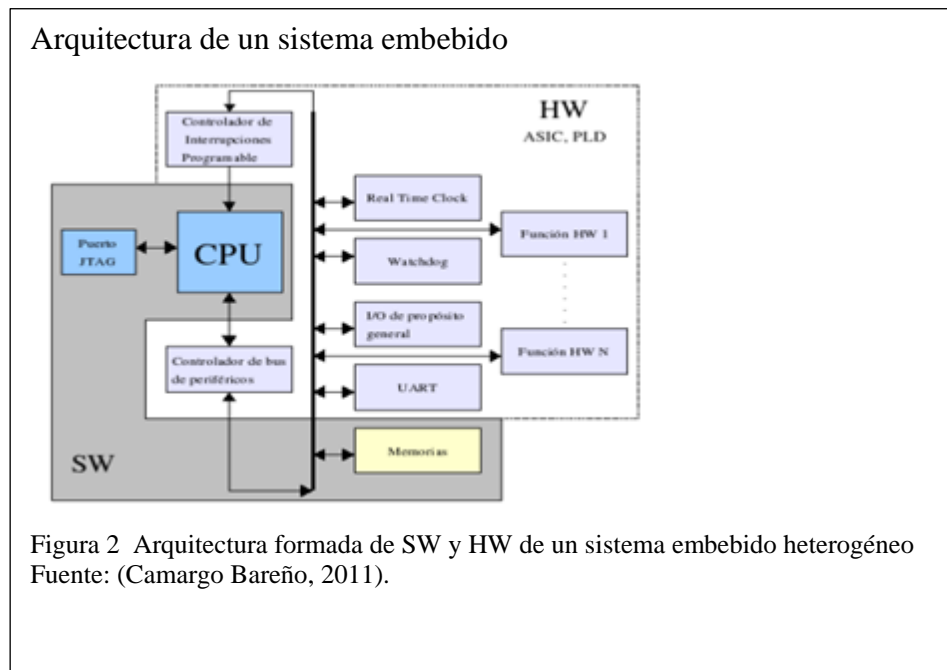
- JTAG: Es una interfaz utilizada para probar circuitos electrónicos, que permite garantizar su funcionamiento y ayuda en la detección de fallas. Actualmente se usa en la depuración de sistemas embebidos.
- RTOS: Como expresa (Aroca et al., 2009), un sistema operativo en tiempo real está diseñado para cumplir con rigurosas limitaciones de tiempo, característica principal de los sistemas embebidos. Brindan un tiempo de cambio de contexto que se ajusta a las necesidades del sistema; es decir permiten guardar el contexto de la tarea anterior y atribuir el contexto de la nueva tarea, además permiten asegurar una latencia al completar una instrucción. (Pérez, 2003)

2.1.2.2 Componentes hardware

- Procesador: Ejecuta las tareas de software y provee rutas de acceso a los periféricos.
- Memorias: Aquí se encuentra la memoria volátil que almacena las variables y la estructura de datos para la ejecución de programas y la memoria no volátil que permite almacenar los datos y programas.
- Periféricos: Esta el controlador de interrupciones programable, interfaces de comunicaciones seriales, temporizadores.
- UART: Con base en (Díaz Mulas, 2015), un transmisor receptor asíncrono universal es un chip cuyo objetivo es convertir los datos recibidos en forma

paralela a serie; a través de una única línea. Permitiendo una comunicación full-duplex; la transmisión y recepción se realizan de manera simultánea.

- Watchdog: Es un módulo de seguridad que consta de un evento, una condición y una acción. Cuando se detecta un evento se evalúa la condición de este y posteriormente ejecuta una acción es utilizado cuando el sistema se haya bloqueado esto genera un reset. (Murcia Esquivia, 2018)



El programa que ejecuta un sistema embebido generalmente es elaborado en lenguaje ensamblador, C++, Basic o ANSI C, siendo este último el más conocido.

2.1.3 Lenguajes de programación

Como afirma (Arriarán, 2017), antiguamente un sistema embebido se programaba en lenguaje ensamblador conocido como un lenguaje de bajo nivel. Esto obligaba a que el programador tenga conocimientos sobre el hardware y la arquitectura del sistema embebido, lo cual lo convertía en un lenguaje complejo y su tiempo de programación era muy alto, además sus instrucciones dependían de la marca y la arquitectura.

En 1972 se desarrolla el lenguaje C creado por Dennis Ritchie y Brian Kernighan en los laboratorios Bell, el cual aun siendo un lenguaje de alto nivel por sus características aparentaba ser de bajo nivel permitiendo acoplarse a un lenguaje de máquina. En 1988 el Instituto Nacional Estadounidense de Estándares (ANSI) designa al lenguaje C como un estándar dando paso al lenguaje ANSI C, siendo el más utilizado actualmente para la programación de sistemas embebidos. (Arriarán, 2017)

2.1.4 Lenguaje ANSI C

(Russell, 2010) menciona que el lenguaje ANSI C define un conjunto de reglas que rigen la sintaxis, los tipos y las llamadas a funciones, está en el núcleo de cualquier compilador C. Posee un código portable es decir independientemente de la plataforma del hardware, si el software está escrito en ANSI C podrá usarse para crear un programa ejecutable en cualquier procesador. Esto ayudo a solventar algunos problemas presentados en el lenguaje ensamblador como es la compatibilidad, a diferencia del lenguaje ensamblador sus instrucciones son las mismas sin depender de la marca ni la arquitectura.

(Arriarán, 2017) describió las ventajas del lenguaje ANSI C de la siguiente manera:

- Facilita la generación de código de mediana y alta complejidad
- El código es interpretable para cualquier persona con conocimientos en C y sistemas embebidos
- Permite crear librerías compartidas en la comunidad de programadores

Por otro lado, no es recomendable usar cuando el sistema embebido tiene recursos de memoria limitados, ya que al ser un lenguaje de alto nivel requiere espacios de almacenamiento para determinados tipos de variables. Actualmente, ANSI C es muy utilizado por los programadores de sistemas embebidos. A continuación, se describe las aplicaciones de estos sistemas.

2.1.5 Aplicaciones

De acuerdo con (Camargo Bareño, 2011), los sistemas embebidos están en casi todas las actividades cotidianas de un ser humano y se estaría interactuando a diario con estos sistemas, pasándolos desapercibidos. Estos sistemas han sido utilizados en diversos campos; medicina, sistemas de control, mecánica, industria automotriz, domótica (Hongxing & Tianmiao, 2006). En la Tabla 1 se describe algunos estudios en los que se utilizan sistemas embebidos como base en su desarrollo, mejorando el estilo de vida y automatizando tareas que hasta hace unos años parecía imposible.

(Chyliński & Szmajda, 2019) propone un sistema para realizar un examen pulmonar, específicamente medir la frecuencia respiratoria, y un software para PC para analizar la señal de dicha frecuencia. En cuanto (Ye, 2020), da a conocer un sistema integrado de monitoreo de la red de tuberías para una ciudad inteligente que permite realizar consulta de datos históricos e implementa funciones de monitoreo y alarma en tiempo real para los parámetros de estado, por ejemplo, caída de temperatura, caída de presión en cada sección en la red de tuberías. Algo similar sucede en el campo de domótica que utiliza estos sistemas para realizar varios proyectos juntamente con el internet de las cosas.

Tabla 1 Campos y aplicaciones que han utilizado sistemas embebidos

Campo	Aplicación
Medicina	Sistema embebido que ayuda a medir la frecuencia respiratoria para diagnosticar una enfermedad del sistema respiratorio. (Chyliński & Szmajda, 2019)
Mecánica	Control de flotas en tiempo real. (Vitores, 2004)
Domótica	Los sistemas embebidos juntamente con IoT han permitido realizar varios proyectos.
Robótica	Sistema de monitoreo de red de tuberías basado en un sistema embebido para analizar sus nodos y detectar fallas. (Ye, 2020)

Nota. Esta tabla muestra algunos estudios que usaron sistemas embebidos para su realización

Elaborado por: Los Autores

Para que estas aplicaciones sean posibles se usan placas electrónicas que permiten una variedad de posibilidades de comunicación en un espacio reducido. Entre estas la más utilizada es Arduino que será detallada a continuación.

2.2 Arduino

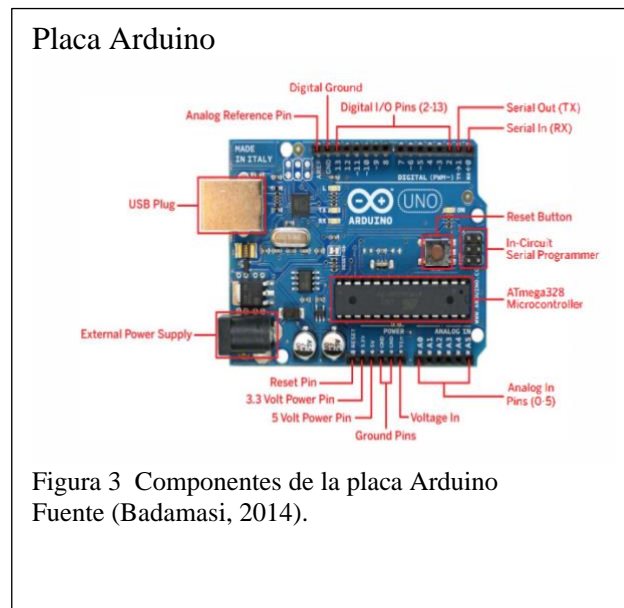
Es una plataforma basada en hardware y software libre, compuesta por un microcontrolador, sensores y un conjunto de pines para crear conexiones entre el microcontrolador y dichos sensores, es de código libre y cuenta con una comunidad que permite compartir código entre programadores. (Arduino.cl, s.f.)

Las instrucciones se escriben en un software IDE, que es un programa que se ejecuta en cualquier computadora con un intérprete de Java y permite escribir y compilar el código. (Russell, 2010)

En la Figura 3 se puede identificar los componentes que forman parte de la placa Arduino. De acuerdo con el esquema presentado por (Badamasi, 2014), Arduino está compuesto por:

- Fuente de alimentación externa: Se usa para alimentar la placa con un voltaje de 9 a 12 voltios.
- Enchufe USB: Es utilizado para cargar un programa al microcontrolador, con una potencia de 5 voltios que permite alimentar a la placa, no tan eficiente como la fuente de alimentación externa.
- Microcontrolador ATmega328: Envía y recibe información a través de la ejecución de instrucciones.
- Pines de entrada/salida: Son pines que permiten la entrada y salida digital.
- Pines de alimentación: Cuenta con una potencia de 3,3 y 5 voltios
- Botones de inicio, reseteo: Permite encender la placa y restablecer el Arduino, en caso de ser necesario.

- Pines analógicos: Son pines de entrada analógica de AO a A5.
- Pines de tierra: Ayuda con la polarización del circuito.



Arduino cuenta con diversos modelos de placas, cada una de ellas tiene diferentes características y fue creada con un propósito específico. Sin embargo, todas estas placas pertenecen a la misma familia de microcontroladores y su arquitectura, documentación y librerías son similares.

2.2.1 Placas Arduino

En la Tabla 2 se describe algunas características técnicas de las placas de Arduino más usuales, utilizadas para numerosos proyectos.

Tabla 2 Placas Arduino

Placas Arduino	Ethernet	Arduino DUE	ADK	Wi-Fi REV2
Microcontrolador	ATmega 328	AT91SAM3X8E	Atmega2560	ATmega4809
Flash Memory	32 KB	256 KB	256 KB	48 KB
Pines digitales E/S	14	54	54	14

Nota. Tabla comparativa de placas de Arduino más utilizadas

Fuente (paraarduino, 2020)

La placa Ethernet permite conectividad a través de un controlador TCP/IP integrado en la placa, permitiendo transferir datos entre cualquier dispositivo conectado a la red. Wi-Fi REV2 provee conectividad Wi-Fi a los dispositivos adheridos a esta placa. Arduino ADK está diseñado para interactuar con teléfonos móviles de sistema operativo Android (Artero, 2013). Para establecer una comunicación con las placas Arduino es necesario realizar una serie de instrucciones a través del entorno IDE conocidas como comandos AT, detallados a continuación.

2.2.2 Comandos AT

(Velasco, 2005) señala que, en 1977 Dennis Hayes desarrolló el juego de comandos AT como una interfaz de comunicación con un módem, éste permitía que sea configurado y recepte instrucciones específicas, por ejemplo; marcar un número telefónico. Posteriormente compañías como Microcomm y US Robotics se encargaron de ampliar y propagar el juego de comandos hasta lograr que se extienda.

Los comandos AT cuya denominación proviene de abreviar la palabra attention, son también conocidos como Hayes; éstos conforman un lenguaje de instrucciones codificadas que permiten la comunicación entre el hombre y un terminal módem. Además, fueron creados para que la compatibilidad entre módems diferentes esté garantizada. (Velasco, 2005)

Como señala (Bonillo, 2004), el éxito que obtuvieron dichos comandos les permitió convertirse en un estándar de comunicaciones virtuales. Asimismo, de los módems modernos el 99,99% comprenden estos comandos, por lo que son denominados compatibles para Hayes.

Los comandos AT tienen como finalidad principal mantener una comunicación directa con los módems GSM, todos estos poseen un conjunto de comandos específicos, los cuales a través de una interfaz de configuración conocida como RS232,

permiten enviar instrucciones a sus diferentes terminales; toda esta interacción está basada en código ASCII, lo que permite que desde cualquier ordenador se pueda acceder al modem, sea para configurarlo, actualizarlo, hacer pruebas o para conectarlo a otro modem. (Velasco, 2005)

Dentro de los comandos AT, se describen dos tipos principales: comandos que ejecutan acciones inmediatas y comandos que cambian algún parámetro del módem (Bonillo, 2004). Un claro ejemplo que involucra lo mencionado anteriormente es la telefonía móvil GSM, la misma que ha implementado este tipo de lenguaje como un estándar principal en la comunicación entre sus diferentes dispositivos. Esto ha permitido realizar acciones como: llamadas de voz y envío de SMS, además de diversos tipos de configuración dentro del dispositivo. (Velasco, 2005)

Todas las configuraciones que se realizan en cada uno de los dispositivos mencionados se detallan en la Tabla 3, la cual contiene los comandos principales para el funcionamiento de cada uno de los módulos.

Tabla 3 Comandos AT

COMANDOS	DESCRIPCIÓN
AT	Indica el funcionamiento y arranque del dispositivo
AT+CGMI	Nos muestra todas las características principales del fabricante
ATI	Permite ver la información del dispositivo como el nombre, código y versión del hardware.
AT+IPR=XXX	Permite configurar la frecuencia de comunicación entre la página de inicio y el módulo.

AT+COPS?	Muestra el nombre de la compañía de telecomunicaciones a la cual el módulo se encuentra anclada.
AT+CMGL=ALL	Sirve para ver todos los mensajes que han llegado al dispositivo, solo si el mismo cuenta con una tarjeta SIM instalada.
AT+CGSN	Permite ver el IMEI de la tarjeta SIM del dispositivo.

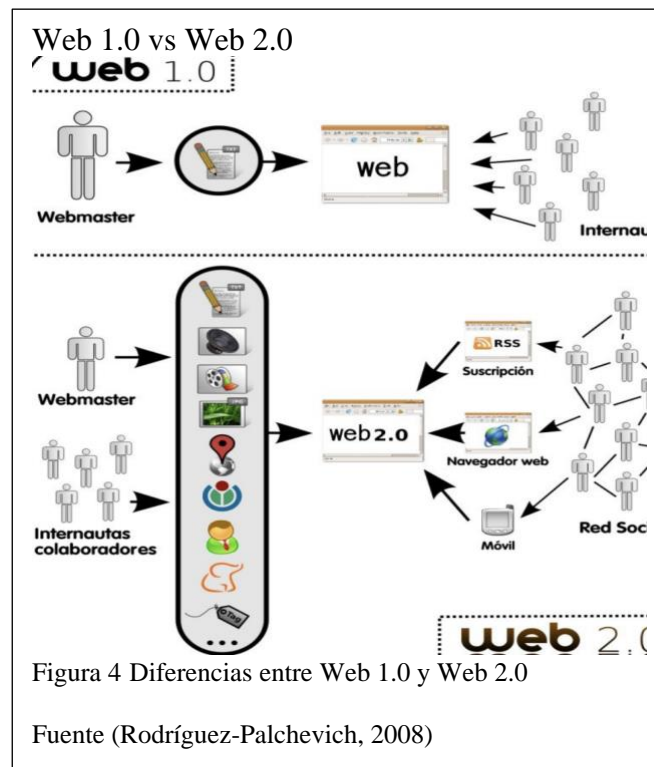
Nota. Tabla descriptiva de los comandos AT más utilizados
Fuente (Velasco, 2005)

Como ya se ha mencionado anteriormente, todos estos comandos permiten realizar configuraciones determinadas en sus dispositivos, dichas configuraciones son realizadas de forma local mediante el uso de las tecnologías Web, las mismas que serán descritas en el siguiente punto.

2.3 Tecnologías Web

Como plantea (Rodríguez-Palchevich, 2008), a lo largo del tiempo se han desarrollado diferentes modelos que se basan en una tecnología Web, debido a que la variedad de usos que proporcionan representa una herramienta indispensable para colaborar en el desarrollo de empresas o negocios. Las tecnologías Web pasaron de ser una simple fuente de información a convertirse en una herramienta de trabajo colaborativa que le permite al usuario interactuar, realizar publicaciones y escribir contenidos como se indica en la Figura 4. En sí, el uso de estas tecnologías ha permitido que la mayoría de las empresas adapten sus diferentes aplicaciones y modelos de trabajo a una plataforma Web, consiguiendo de esta manera darse a conocer e incrementar la productividad en su modelo de negocio. Por tal motivo, las tecnologías Web han implementado diferentes características y herramientas, con el

fin de hacer de su uso algo fácil e intuitivo, logrando así que el manejo de información en la Web pase de una red centralizada a una red robustamente distribuida.

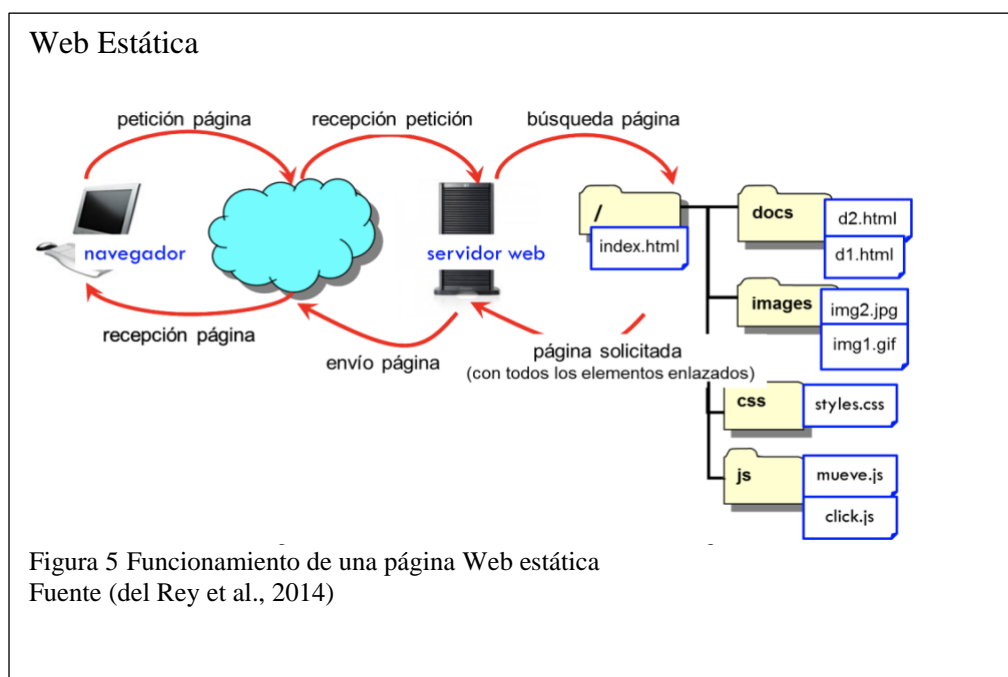


A continuación, se describen los tipos de tecnologías Web que se usan en la actualidad al igual que su arquitectura base.

2.3.1 Páginas Web Estáticas

(del Rey et al., 2014) considera que una página Web estática es una página que se enfoca principalmente en mostrar información de manera permanente, por lo que el navegante únicamente puede obtener la información, pero no tiene la facilidad de interactuar con la página Web que está visitando. Este tipo de páginas no necesitan numerosas actualizaciones de manera constante. Están formadas únicamente por código HTML y un estilo (CSS). Además, están construidas mediante hipervínculos o enlaces con las demás páginas Web que forman parte de un sitio. Por otra parte, estas páginas Web, no tienen la capacidad de tolerar aplicaciones Web tales como gestores de bases de datos, consultas on-line, foros o e-mails inteligentes.

Son diseñadas principalmente para micro o macroempresas, que buscan únicamente que su sitio Web sea informativo, es decir, que quieran dar a conocer su perfil para que el usuario lo pueda ver, pero no interactuar. Las páginas estáticas son entrelazadas de formas diferentes, las cuales hacen referencia a elementos dependientes; cualquier petición que realice el usuario. Este tipo de páginas se limitará a buscar entre sí como se muestra en la Figura 5. Este tipo de estructura es de fácil manejo y no muy complicada al momento de desarrollarla. (del Rey et al., 2014)



2.3.2 Páginas Web Dinámicas

Las páginas Web dinámicas permiten alojar diferentes tipos de servicios o aplicaciones dentro de la Web, lo que hace posible que exista una mayor interacción entre la página y el navegante. Este tipo de páginas tienen un desarrollo más complejo, debido a que necesitan determinados conocimientos acerca de lenguajes de programación y gestión de base de datos. Sin embargo, los resultados obtenidos demuestran que la inversión y el esfuerzo que se invierte valen la pena, por la gran potencia y el servicio que éstas ofrecen. Estas páginas Web dinámicas se basan en la

tecnología WEB 2.0 (del Rey et al., 2014), a continuación se detallara la arquitectura de dichas páginas.

2.3.3 Arquitectura básica de una aplicación Web

Los sitios Web han tenido una notable evolución, en sus inicios estos se limitaban únicamente a mostrar información; sin embargo, actualmente, gracias al incremento progresivo en su complejidad, son capaces de permitir que exista una mayor interacción con el usuario. Este hecho hace pertinente indagar nuevas opciones de diseño que permitan construir aplicaciones con una arquitectura idónea, lo que contribuiría a la optimización de sus servicios.

El usuario es el que mediante el navegador puede interactuar con las aplicaciones Web, como consecuencia el usuario envía peticiones que se dirigen al servidor, en donde se encuentra alojada la aplicación, la cual hace uso de una base de datos. En sí, el servidor se encarga de procesar la petición emitida por el usuario, para posteriormente presentar una respuesta al navegador, dicha respuesta, a su vez, será mostrada al interesado. Se puede concluir que son tres componentes en los que este sistema se distribuye, así tenemos: el navegador, que es el encargado de presentar la interfaz al usuario; la aplicación Web, responsable de llevar a cabo operaciones que son requeridas de acuerdo con las acciones que ejecute el interesado; y, la base de datos, en donde toda la información solicitada se apila y es estructurada para al fin ser enviada y visualizada por el usuario. A esta distribución se la conoce como el modelo de tres capas (Almaraz Hernández et al., 2011).

El navegador, por lo general suele únicamente presentar la información sin llevar a cabo procesamientos que estén relacionados con la lógica de negocio, es decir el caso de un modelo de cliente delgado. Sin embargo, debido al empleo de applets de código JavaScript y de DHTML, los sistemas en su mayoría se ubican entre un modelo

de cliente delgado y uno de cliente grueso, es decir el cliente o usuario se encarga de procesar la información, mientras que el servidor únicamente se responsabiliza de administrar los datos. Cabe mencionar que el procesamiento se relacionaría con aspectos de la interfaz mas no con la lógica de negocios. (Vignaga & Perovich, 2008)

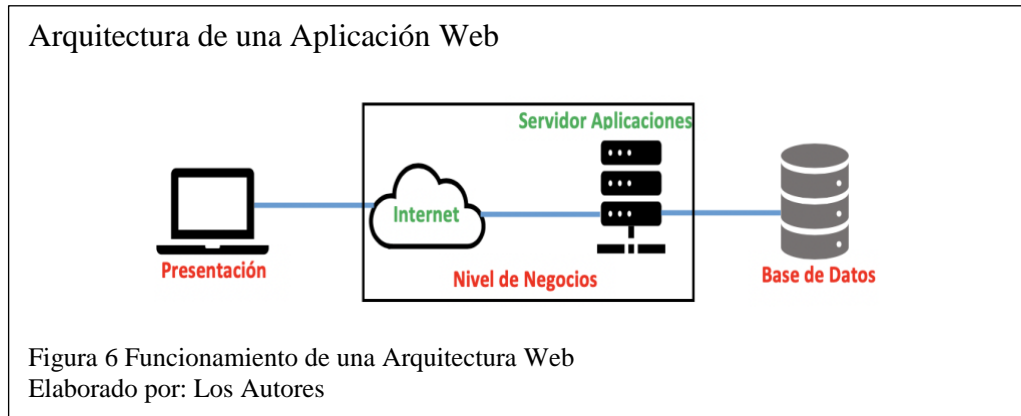
En este tipo de sistemas la aplicación se puede dividir en tres niveles como se muestra en la Figura 6:

- a) Nivel de presentación
- b) Nivel de negocio
- c) Nivel de administración de datos

El nivel de presentación se encarga de crear la interfaz de usuario en relación con las acciones que son ejecutadas por el cliente.

Para (Almaraz Hernández et al., 2011), el nivel de negocio posee los métodos que se encargan de modelar los procesos de negocio; además, en este nivel se lleva a cabo el procesamiento demandado para poder atender los requerimientos que plantea el usuario.

Por último, el nivel de administración de datos se ocupa de hacer persistente la totalidad de información, también se encarga de proveer y recopilar información para el nivel de negocio. Tanto el nivel de presentación como el de negocio acompañados de una parte del nivel de administración, se localizan en el servidor. La parte que resta del tercer nivel está situada en la base de datos. (Almaraz Hernández et al., 2011)

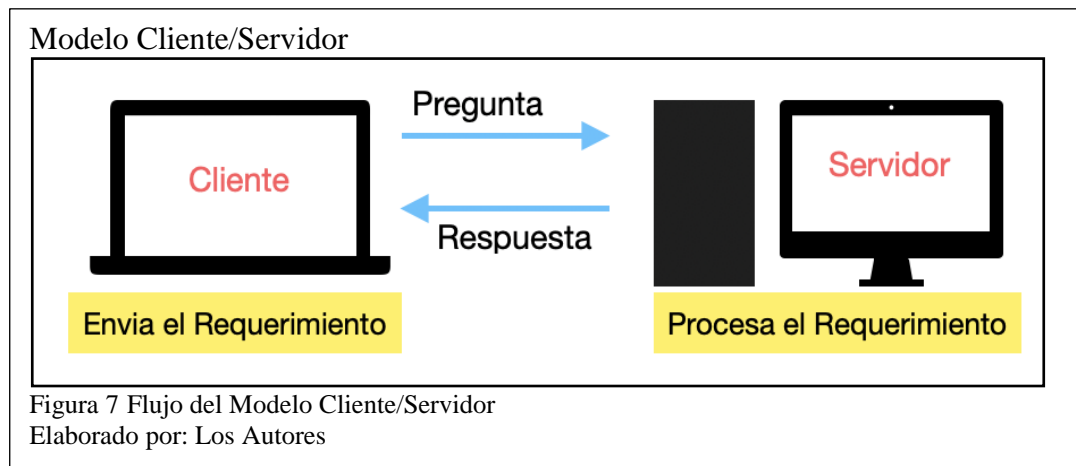


2.3.4 El Modelo Cliente / Servidor

El modelo Cliente/Servidor es un sistema distribuido, el mismo que permite que los usuarios finales puedan acceder a la información de manera transparente, incluso en entornos multiplataforma. Este modelo permite que el cliente realice una petición mediante el envío de un mensaje, en el que solicita un determinado servicio al servidor. Este último una vez recibida la petición, enviará una respuesta a la solicitud inicial como se muestra en la Figura 7. En este tipo de arquitectura distribuida, la principal característica es que cada dispositivo cumple el rol de servidor para ciertas actividades y el rol de cliente para otras. (Marini, 2012)

En sí, el modelo Cliente/Servidor presenta varias ventajas, entre ellas la presencia de plataformas de hardware más económicas; la facilidad de permitir que sistemas diferentes se integren, además de compartir información, lo que facilita la utilización de interfaces más amigables para el usuario. Su mantenimiento es más rápido al igual que el desarrollo de aplicaciones, sin embargo, a pesar de contar con muchas ventajas este modelo también presenta ciertas desventajas tales como: mayor dificultad en el mantenimiento, escasez de herramientas para administrar y ajustar su desempeño.

A continuación, se da a conocer como este modelo funciona de forma individual tanto el cliente como el servidor.



2.3.5 Cliente

Las aplicaciones Cliente se definen como el proceso que le permite al usuario expresar las peticiones requeridas y transferirlas al servidor, dicho proceso es conocido como front-end. El cliente es quien inicia la exigencia de un servicio, sin embargo, esta exigencia inicial podría transformarse en múltiples peticiones de trabajo, por medio de redes LAN o WAN.

Tanto la ubicación de estas aplicaciones, así como de la información son transparentes para el cliente. (Márquez Avendaño & Zulaica Rugarcía, 2004)

Estas aplicaciones llevan a cabo ciertas funciones, entre ellas:

- Recibir los resultados del servidor.
- Interactuar con el usuario.
- Recopilar información emitida por el usuario.
- Depurar resultados.

2.3.6 Servidor

El servidor se encarga de responder las peticiones de diversos clientes, este proceso es conocido como back-end, se puede conectar con los clientes mediante una red LAN (Local Area Network) o una WAN (Wide Area Network), lo que permite que al usuario se lo provea de una variedad de servicios. Por lo general, el servidor es quien maneja las funciones que se relacionan con las reglas del negocio y los recursos de datos. (Márquez Avendaño & Zulaica Rugarcía, 2004)

Entre las funciones principales que tienen los Servidores están:

- Aceptar o denegar los requerimientos enviados por el usuario.
- Depurar requerimientos.
- Búsqueda de requerimientos en la Base de Datos.
- Validar y enviar requerimientos solicitados.

Este tipo de modelos son encapsulados en lenguajes de programación los cuales permiten tener una comunicación bidireccional, descritos a continuación.

2.4 Lenguajes de Programación Web

Los lenguajes de Programación son esquemas organizados de reglas las cuales forman algoritmos dentro del ordenador y que les permiten comunicar ideas lógicas al usuario, este definido por una sintaxis, semántica y excepciones; las características principales de este tipo de lenguajes es que permiten ser portables, fáciles de comprender y con una mayor visión para resolver problemas. (García et al., 2005)

A continuación, se describen los lenguajes más utilizados para la creación de una página Web tanto dinámica como estática.

2.4.1 Lenguaje HTML

(Ferrer et al., 2013) argumenta que HTML es un lenguaje de programación el cual es entendido y procesado por la máquina para que posteriormente ésta pueda emitir una respuesta, sus siglas Hyper Text Markup Language, cuya traducción al español es lenguaje de marcas de hipertexto. En una computadora el hablar de hipertexto, hace relación a un texto que tiene referencias a otro texto mediante hipervínculos, enlaces o links; es decir, cuando se coloca el mouse sobre un determinado texto, éste nos conducirá a otro texto al momento que se utilice internet.

Cabe mencionar que el hipertexto aparte de texto puede estar conformado por otros elementos tales como gráficos, tablas, entre otros. A través de su uso se ha podido determinar dos grandes ventajas que hacen de este lenguaje una de las herramientas más indispensables al momento de crear una página Web, su compatibilidad y la facilidad que proyecta su enseñanza puesto que se apoya en un número reducido de etiquetas. (Vértice, 2009)

2.4.2 Lenguaje CSS

CSS significa Cascading Style Sheets, que traducido al español es hojas de estilo en cascada. Es una herramienta de estilo, ésta se emplea para dar estilo a documentos tanto HTML como XML, permitiendo separar lo visual del contenido. En sí, los estilos van a definir la manera en la que se muestran los elementos. (Herrador, 2010)

En este caso, si se realiza algún cambio en el estilo de un elemento en específico en CSS, se extenderá a las páginas con las que se vincule y que, a su vez aparezca dicho elemento; con esto se logra controlar varias páginas Web de manera simultánea, tanto en el formato como en el estilo. Como expresa (Herrador, 2010), CSS funciona

basándose en reglas, las mismas que están conformadas por dos partes: el selector y la declaración, que a su vez está formada por una propiedad y por un valor que se asigna. El selector actúa como un conector entre el documento y el estilo, detallando los elementos que se afectarán por dicha declaración, mientras que la segunda parte establece qué efecto se elegirá. (Herrador, 2010)

2.4.3 Lenguaje JavaScript

Citando a (Mohedano et al., 2012), JavaScript fue creado con la necesidad de realizar páginas dinámicas e interactivas con el usuario, la diferencia que existe entre una página Web dinámica 2.0 con una Web dinámica desarrollada con JavaScript, es que las primeras son dinámicas desde su servidor y vistas por el usuario donde sus acciones permitan realizar cambios directamente, mientras que las creadas con JavaScript permiten que las páginas sean dinámicas desde la pantalla de usuario, este proceso permite una interacción entre la máquina y el beneficiario; en otras palabras, todo el proceso que realice el usuario en la página no afectará a la original, que se encuentra alojada en su servidor. Esto ocurre debido a que JavaScript es un lenguaje interpretado, lo que significa que todas las instrucciones realizadas por el usuario son resueltas por el intérprete de JavaScript, el cual se aloja dentro del navegador permitiendo que el tiempo de respuesta sea inmediato. Es importante mencionar que JavaScript no tienen ninguna similitud con el lenguaje Java por más que sus nombres se asemejen. JavaScript posee varias características, dentro de las más importantes se pueden mencionar las siguientes:

- No necesita compilación debido a que el navegador del usuario es el que interpreta las sentencias JavaScript presentes en una página HTML y puede ejecutarlas de manera apropiada.

- Está direccionado a eventos al momento en que el usuario da clic en un enlace, mueve el puntero sobre una imagen y se origina un evento. Por medio de JavaScript se pueden generar scripts que produzcan una acción como respuesta a dichos eventos.
- Se orienta a objetos, los cuáles son reducidos y simplificados; sin embargo, estos objetos contienen los elementos precisos para permitir el acceso de los scripts a la información de una página y para que actúen sobre la interfaz del navegador. (Mohedano et al., 2012)

Estos lenguajes han permitido que su aplicación en otras áreas crezca de forma exponencial tal es el caso de la medicina, la cual ha permitido que todo lo proporcionado anteriormente se pueda aplicar en fantomas obstétricos. Estos fantomas son utilizados para simulaciones de emergencias médicas, los cuales disminuyen el riesgo de muerte por una mala práctica médica, posteriormente se irán detallando cada uno de ellos.

2.5 Fantoma Obstétrico de Baja Fidelidad

A lo largo de los años los recursos de demostración utilizados en la enseñanza obstetricia han ido evolucionando como es el caso de los fantomas o maniquís obstétricos, su objetivo es simular de una forma verídica el proceso del parto y las complicaciones que se pueden presentar como se muestra en la Figura 8.

En Francia en el siglo XVIII surgió uno de los primeros antecedentes sobre el uso de estos materiales, una matrona llamada Angélique Marguerite Le Boursier du Coudray diseñó este prototipo con el fin de capacitar para atender partos. (Ruiz-Berdún & Blanco, 2014)

Maniqués Obstétricos



Figura 8 Prototipo Obstétrico diseñado en el siglo XVIII
Fuente (Gardner & Raemer, 2008)

El uso de maniqués obstétricos se volvió imprescindible en el transcurso de los años, en el siglo XIX una madrona española escribió el primer libro para la formación de sus alumnas el cual describía el diseño de una pelvis artificial. Actualmente los fantomas obstétricos han evolucionado significativamente (Gardner & Raemer, 2008), refiere que los maniqués se han convertido en simuladores de parto complejos y sofisticados que facilitan la enseñanza de esta profesión milenaria.

A continuación, se analizará uno de los modelos más conocidos, empleados en los fantomas obstétricos en la actualidad:

2.5.1 Modelo de palpación abdominal

El modelo de palpación abdominal representa el abdomen de una mujer en el cual se pueden determinar la posición del feto, comprobar la muerte fetal y realizar la maniobra de Leopold.(Carvajal & Ralph, 2017)

Modelo palpación abdominal



Figura 9 Modelo abdominal de palpación
Fuente (Medical Expo, 2020)

2.6 Simulación de emergencias obstétricas

Es un método utilizado para practicar como se debería manejar una emergencia en el caso de presentarse, su objetivo principal mejorar la calidad y seguridad de la atención para mujeres y recién nacidos. La simulación obstétrica es una representación clínica que involucra una mujer embarazada o que ya dio a luz y se examina e investiga su comportamiento. (Gardner & Raemer, 2008)

2.6.1 Escenarios del uso de un Fantoma Obstétrico de Baja Fidelidad

La simulación ginecológica avanzado considerablemente con el uso de simuladores que exploten las habilidades del estudiante, se pueden trabajar sus destrezas obteniendo resultados en un ambiente casi real. (Greif et al., 2015)

Existe una técnica denominada “Simuladores híbridos” utilizada para simular de manera más realista un entorno clínico, dichas técnicas son especialmente útiles cuando los recursos o las capacidades de almacenamiento son limitados teniendo en cuenta la portabilidad tal como se indica en la Figura 10. (Gardner & Raemer, 2008)

Simulador obstétrico híbrido



Figura 10 Ejemplo de una técnica híbrida que combina un simulador obstétrico de baja fidelidad con un actor.

Fuente (Gardner & Raemer, 2008)

2.6.1.1 Hemorragia postparto

La hemorragia posparto es la pérdida de sangre mayor a 500 ml después de un parto o mayor a 1000 ml después de una cesárea en las 24 horas posteriores al nacimiento (Solari et al., 2014) . Se clasifica en:

- Primaria: Es aquella que ocurre dentro de las primeras 24 horas postparto
- Secundaria: Es aquella que ocurre entre las 24 horas y las 6 semanas postparto

2.6.1.2 Preeclampsia

Se trata de una complicación en el embarazo que se puede presentar en el periodo de gestación, caracterizada por la presencia de presión arterial alta y algunos síntomas como fuertes dolores de cabeza de gran riesgo que pueden afectar la vida del bebe incluso de la madre. (Carty et al., 2010)

2.6.1.3 Eclampsia

La eclampsia en el periodo de gestación es un padecimiento hipertensivo después de la semana 20 del embarazo o las horas posteriores del parto. Así también se denominan a las convulsiones que pueden llegar a ser un riesgo vital y pueden ocurrir antes, durante o después del parto. (Ramoneda & Mussons, 2008)

2.6.1.4 Maniobras de Leopold

Son un conjunto de movimientos que realizan los ginecólogos sobre el abdomen de las mujeres en periodo de gestación para identificar la posición del bebe. (Carvajal & Ralph, 2017) refiere que realizan 4 tipos de maniobras:

- Identificar la presentación fetal.
- Determinar la posición fetal.
- Identificar cual polo fetal ocupa la parte inferior del abdomen materno.
- Determinar la actitud fetal

CAPÍTULO III

ALCANCE DEL PROYECTO

Se construirá un prototipo electrónico que permita controlar y programar simulaciones de diferentes escenarios clínicos. Este prototipo utilizará un módulo Wi-Fi permitiendo lograr una comunicación estable y segura, en la cual se desarrollarán todos los parámetros de conexión y seguridad. Además, se construirá una plataforma Web que brindará al usuario un fácil acceso, y un correcto funcionamiento del fantoma.

La plataforma Web proporcionará al usuario el control total sobre el simulador, el docente podrá controlar los diferentes movimientos y síntomas como: ojos, pulsos cardíacos, sonidos pulmonares, sonidos del corazón y la bomba de sangrado. Este tipo de síntomas tienen que ser reflejados de forma inmediata en el simulador, esto se logrará gracias a los beneficios que posee el módulo a utilizar.

Una vez que se haya establecido el prototipo es necesario realizar un análisis de factibilidad, el cual determinará si el desarrollo del proyecto es abordable.

3.1 Análisis de factibilidad

Se realizará un estudio de factibilidad el cual ayudará a identificar si el proyecto es viable y si se lograra cumplir con los objetivos planteados, para esto se analizará la factibilidad técnica y operativa del proyecto.

3.1.1 Factibilidad Técnica

Para analizar la factibilidad técnica, se debe tener en cuenta todos los recursos necesarios para realizar el prototipo, tanto en hardware como en software.

3.1.1.1 Hardware

Es necesario identificar las herramientas con las que se debe trabajar, al igual que los recursos mínimos que estas necesitan para un correcto desarrollo, evitando futuros inconvenientes e incidencias. A continuación, se detallarán las especificaciones necesarias para el desarrollo del prototipo. Se necesita una PC que cuente con las siguientes características mínimas como se muestra en la Tabla 4.

Tabla 4 Especificaciones mínimas de una PC

ESPECIFICACIONES MÍNIMAS DE UNA PC	
Procesador	Intel Core i5
Sistema Operativo	Windows 10 Home
Memoria RAM	8Gb
Disco Duro	500GB
Tarjeta de Video	4GB

Nota. Esta tabla muestra las especificaciones usadas para el desarrollo de la solución

Elaborado por: Los Autores

Otra de las consideraciones que se debe evaluar son las especificaciones que actualmente se encuentran en el simulador el cual utiliza un módulo Bluetooth HC-05 como un medio de transmisión y recepción, siendo objeto del presente estudio. A continuación, se detallarán sus características en la Tabla 5.

Tabla 5 Especificaciones del módulo Bluetooth

ESPECIFICACIONES DEL MÓDULO BLUETOOTH	
Voltaje	3.6V – 6V DC
Frecuencia	2.4GHz
Potencia de Transmisión	4dBm
Protocolo Soportado	802.15.1
Alcance	10 metros

Nota. Esta tabla indica las especificaciones principales que usa el módulo Bluetooth en el simulador

Elaborado por: Los Autores

Teniendo en cuenta las especificaciones mencionadas anteriormente, se realizará una descripción de las características más importantes del módulo Wi-Fi como se especifica en la Tabla 6, en la cual se podrá evidenciar una mejora considerable tanto en la transmisión como en el alcance a diferencia del módulo bluetooth.

Tabla 6 Especificaciones del módulo Wi-Fi

ESPECIFICACIONES DEL MÓDULO WI-FI ESP8266 – CP2102	
Voltaje	5V DC 3.3V DC (Voltaje lógico in/out)
Soporte de Red	2.4 GHz
Potencia de Transmisión	>19.5dBm
Protocolo Soportado	802.11 b, g, n
Alcance	30m de alcance teórico

Nota. Esta tabla muestra las especificaciones principales que usa el módulo Wi-Fi en el simulador

Elaborado por: Los Autores

3.1.1.2 Software

Una vez que se identificaron los dispositivos que se utilizarán es necesario enfocarse en todas las herramientas complementarias que serán vitales para el desarrollo de la solución.

Una de las herramientas que ayudarán en la construcción de la página Web será el lenguaje HTML, este lenguaje a más de ser escalable posee una amplia documentación la cual ayudará a solventar los diferentes problemas que se presenten a lo largo de la construcción del sistema, una de las características principales de este lenguaje es su interacción con la herramienta CSS, la misma que cuenta con una extensa gama de etiquetas que ayudarán a dar un diseño amigable a la página Web.

Para la comunicación con el simulador y la página se hará uso de la herramienta Arduino IDE, la misma que posee un ambiente de desarrollo que logrará integrar la página Web dentro del módulo, de esta manera se podrá trabajar con los datos que la página envíe y usarlos para controlar del simulador.

3.1.2 Factibilidad Operativa

Se contará con un sistema el cual permita al docente ingresar a la página Web a través de una IP desde el navegador, dicha página interactúa con el simulador mediante un módulo Wi-Fi ESP8266 - CP2102 estableciendo la comunicación de datos, comportándose como un servidor.

En la página Web se encuentran varias acciones que el usuario podrá elegir y que se verán reflejadas en el fantoma, estas acciones serán detalladas en la planificación de requerimientos.

Para que los objetivos del proyecto sean culminados con éxito se debe de contar con los recursos humanos necesarios, detallados a continuación.

3.1.2.1 Recursos Humanos

Para el desarrollo del proyecto es necesario contar con el siguiente personal:

Tabla 7 Recursos Humanos

N°	Recurso Humano	Cantidad	Función	Descripción
1	Desarrollador Junior	1	<ul style="list-style-type: none"> • Creación de la página Web (Back End). • Diseño de las interfaces (Front End) 	<ul style="list-style-type: none"> • Encargados de desarrollar y brindar mantenimiento a la página Web
3	Técnico de telecomunicaciones	1	<ul style="list-style-type: none"> • Establecer la comunicación de datos ente la 	<ul style="list-style-type: none"> • Encargado de trabajar a la par con el desarrollador para lograr una

			<p>página y el IDE de Arduino.</p> <ul style="list-style-type: none"> • Identificar las conexiones de los dispositivos del simulador • Crear un diseño lógico para la conectividad entre placas. 	<p>comunicación entre la página y el simulador.</p> <ul style="list-style-type: none"> • De igual manera deberá realizar un estudio minucioso sobre todos los dispositivos conectados en el simulador. • Una vez lograda la comunicación es necesario que se realice un diseño lógico de la placa a la cual ira conectado el nuevo módulo.
4	Doctor encargado de la clínica de simulación	1	<ul style="list-style-type: none"> • Validar el funcionamiento correcto del simulador. 	<ul style="list-style-type: none"> • Encargado de validar cada uno de los funcionamientos del simulador como: <ul style="list-style-type: none"> • Pulsos • Ritmos Cardíacos • Movimientos

Nota: Esta tabla muestra los recursos humanos que se necesitan para el proyecto

Elaborado por: Los Autores

3.2 Resultados esperados

Se espera que el cambio del módulo Bluetooth al módulo Wi-Fi CP2102 ayude a mejorar la comunicación y el tiempo de respuesta del simulador al intentar realizar alguna acción en específico, esto permitirá que las practicas dictadas por el docente sean eficientes. Además, no se tendrá que realizar ninguna configuración por parte del usuario a diferencia del módulo actual, donde se debía instalar el software

correspondiente y a su vez configurar un puerto virtual de comunicación, el cual permitía establecer el control entre la PC y el simulador.

De igual manera se espera que todos los cambios realizados optimicen los recursos de la clínica de simulación y el tiempo en el que el docente realiza la práctica junto a los estudiantes mejorando la calidad de servicio (QoS).

Para cumplir con los objetivos propuestos se debe implementar una metodología que ayude a tener un control sobre el proyecto y optimice el tiempo de su desarrollo, a continuación, se puntualizará dicha metodología.

CAPÍTULO IV

PLANIFICACIÓN

La fase de planificación ayudará identificar los requerimientos del cliente permitiendo tener una retroalimentación continua, una característica principal de esta metodología. A continuación, se analizará los requerimientos tanto funcionales como no funcionales, una vez que se entienda las necesidades del cliente se realizará las historias de usuario a las cuales les corresponderá una tarea de ingeniería que será asignada a cada responsable que cumplan un rol en este proyecto.

4.1 Análisis de requerimientos

Mediante el análisis de requerimientos se logrará identificar las necesidades a cubrir para el correcto desarrollo del sistema y del simulador, este análisis permitirá que el desarrollo de la solución ayude a mejorar la comunicación y optimice recursos, para esto se tomará en cuenta algunas especificaciones dictadas por el Ingeniero a cargo de la clínica de simulación basándonos tanto en los requerimientos funcionales como en los no funcionales.

4.1.1 Requerimientos Funcionales

Los requerimientos funcionales abarcan todo el desempeño del sistema al igual que su operabilidad, los cuales se detallarán en la Tabla 8.

Tabla 8 Requerimientos Funcionales.

CÓDIGO	REQUERIMIENTOS FUNCIONALES
1	Para ingresar al sistema es necesario estar conectado a la red de la clínica de simulación
2	El docente ingresará la dirección IP desde su navegador a la cual esta asignada el simulador.
3	El sistema permitirá que el docente cargue sonidos de la paciente según requiera su práctica.

4	El sistema permitirá realizar el control de los ojos del simulador, para esto se tendrá en cuenta 3 opciones (abiertos, cerrados y medios).
5	El sistema tendrá la opción de controlar los sonidos del corazón y pulmones.
6	Dentro de los sonidos del corazón se tendrá una lista de complicaciones médicas que el simulador debe reproducir, tales como: <ul style="list-style-type: none"> • Normal • Taquicardia • Soplo de Austin Filt • Soplo Diastólico • Soplo Holo Sistólico • Comunicación Interauricular • Soplo Septal • Pausa
7	Dentro de los sonidos de los pulmones se tendrá una lista de complicaciones médicas que el simulador debe reproducir, tales como: <ul style="list-style-type: none"> • Normal • Sibilancias (Espiratoria) • Estridor (Inspiratorio) • Crepitantes (Inspiratorio) • Subcrepitantes (Inspiratorio) • Estertores Húmedos (Inspiratorio) • Pausa
8	El sistema permitirá realizar el control de los pulsos cardíacos, el aspecto de los labios al igual que el control de la bomba de sangre.
9	Dentro de los pulsos cardiacos se tendrá una lista de complicaciones médicas que el simulador debe reproducir, tales como: <ul style="list-style-type: none"> • Bradicardia (Alto). • Normal. • Taquicardia (Bajo) • Paro
10	Dentro del aspecto de los labios se tendrá 2 complicaciones médicas orientadas al escenario obstétrico, tales como: <ul style="list-style-type: none"> • Cianosis (Color Morado) • Labios Normales.
11	Dentro de la bomba de sangre se tendrán en cuenta 2 cambios principales, tales como: <ul style="list-style-type: none"> • Empezar Sangrado. • Detener Sangrado
12	El sistema tendrá la opción de reproducir los sonidos del paciente que se encuentran precargados o los que el docente haya cargado con anterioridad.

Nota. Esta tabla muestra los requerimientos funcionales del sistema

Elaborado por: Los Autores

4.1.2 Requerimientos no Funcionales

Los requerimientos no funcionales se enfocan en la manera en la que el sistema debe funcionar, tanto en su usabilidad como en el rendimiento del mismo, los cuales se detallan en la Tabla 9.

Tabla 9 Requerimientos no funcionales.

CÓDIGO	REQUERIMIENTOS NO FUNCIONALES
1	El sistema contará con guía de usuario.
2	La interfaz de usuario será intuitiva y de fácil manejo para el docente.
3	Se deberá acceder a la página Web desde un ordenador para facilitar su manejo.
4	Se deberá mejorar el tiempo de respuesta entre el simulador y la página Web cuando se realice una acción en específico.
5	El docente será la única persona que manipule las credenciales de accesos para agregar o eliminar sonidos de la página Web según lo requiera.

Nota. Esta tabla muestra los requerimientos no funcionales del sistema

Elaborado por: Los Autores

4.2 Historias de Usuarios

Las historias de usuarios se generarán con el fin de crear un lenguaje común entre los docentes (usuarios) y los desarrolladores, por lo que se detallarán todos los requerimientos del sistema ya mencionados de una forma más clara y específica.

A continuación, en las Tablas 10-16 se muestran las historias de usuarios de forma detallada para el desarrollo del sistema.

Tabla 10 Historia de Usuario Acceso al Sistema

HISTORIA DE USUARIO	
Número: 1	Usuario: Docentes, Ingeniero a cargo
Nombre Historia: Acceso al Sistema	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Media
Puntos Estimados: 2	Interacción Asignada: 1
Programador Responsable: Ángela Román	
<p>Descripción: El docente debe ingresar la IP del simulador desde el ordenador de la clínica de simulación, el ordenador debe estar conectado a la red de la clínica. Si la IP ingresada es correcta se mostrará la ventana principal del simulador.</p>	
<p>Observaciones: Si el docente encargado de la práctica no recuerda la IP del simulador deberá comunicarse con el ingeniero encargado de la clínica de simulación.</p>	

Nota. Esta tabla muestra la historia de usuario al acceder al sistema

Elaborado por: Los Autores

Tabla 11 Historia de Usuario Gestión Multimedia

HISTORIA DE USUARIO	
Número: 2	Usuario: Docentes, Ingeniero a cargo
Nombre Historia: Gestión Multimedia	
Prioridad en Negocio: Baja	Riesgo en Desarrollo: Baja
Puntos Estimados: 1	Interacción Asignada: 1
Programador Responsable: Sergio Palacios	
<p>Descripción: El docente podrá ver en la pantalla principal una barra de menú en la cual al hacer clic en el botón editar encontrará dos ítems; cargar sonidos y credenciales, al hacer clic en el primer ítem le redireccionará a SoundCloud en la cual el docente podrá cargar más sonidos previos a la práctica, para poder realizar esta modificación el docente deberá ingresar las credenciales proporcionadas en el ítem 2.</p>	
<p>Observaciones: Si el docente tiene problemas al ingresar o cargar sonidos deberá comunicarse con el ingeniero encargado de la clínica con anticipación a su práctica.</p>	

Nota. Esta tabla muestra la historia de usuario al gestionar contenido multimedia

Elaborado por: Los Autores

Tabla 12 Historia de Usuario Gestión de movimientos de ojos

HISTORIA DE USUARIO	
Número: 3	Usuario: Docentes, Alumnos
Nombre Historia: Gestión de movimientos de ojos	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Media
Puntos Estimados: 1	Interacción Asignada: 1
Programador Responsable: Sergio Palacios y Ángela Román	
<p>Descripción: El docente podrá modificar los diferentes aspectos de los ojos según la práctica a realizar, estos movimientos pueden ser: abiertos, los cuales representan a una paciente normal; cerrados, los cuales identifican que la paciente está en paro y medios (Obnubilados), los cuales ayudan a identificar que la paciente está sufriendo algún tipo de sangrado grave.</p>	
<p>Observaciones: Si el docente o alumno no visualiza ningún cambio es necesario reiniciar el simulador debido a una pérdida de conexión, esto sucede a que existe inactividad entre la página y el simulador por un tiempo prolongado.</p>	

Nota. Esta tabla muestra la historia de usuario al gestionar los movimientos de los ojos

Elaborado por: Los Autores

Tabla 13 Historia de Usuario Gestión de Sonidos Torácicos

HISTORIA DE USUARIO	
Número: 4	Usuario: Docentes, Ingeniero a cargo, Alumnos
Nombre Historia: Gestión de Sonidos Torácicos	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Medio
Puntos Estimados: 1	Interacción Asignada: 1
Programador Responsable: Sergio Palacios y Ángela Román	
<p>Descripción: El docente podrá controlar los diferentes tipos de sonidos que existen en la cavidad torácica, tanto sonidos del corazón como el de los pulmones.</p> <p>Dentro de los sonidos del corazón el docente tiene una lista de complicaciones que podrá ir adaptando según el escenario que maneje las cuales son: Normal, Taquicardia, Soplo de Austin Filt, Soplo Diastólico, Soplo Holo Sistólico, Comunicación Interauricular, Soplo Septal y Pausa.</p> <p>De igual manera dentro de los sonidos pulmonares el docente tiene una lista de complicaciones que podrá ir adaptando según el escenario que maneje las cuales son: Normal, Sibilancias (Espiratoria), Estridor (Inspiratorio), Crepitantes</p>	

(Inspiratorio), Subcrepitantes (Inspiratorio), Estertores Húmedos (Inspiratorio) y Pausa.
Observaciones: <ul style="list-style-type: none"> • Si el docente o alumno a cargo de la práctica no posee un estetoscopio solicitarlo con anticipación al ingeniero a cargo, ya que sin ayuda del mismo será imposible apreciar los cambios del paciente. • Si el docente o alumno a no presencia ningún cambio con el estetoscopio es necesario reiniciar el simulador debido a una pérdida de conexión, esto sucede a que existe inactividad entre la página y el simulador por un tiempo prolongado.

Nota. Esta tabla muestra la historia de usuario al gestionar los sonidos torácicos

Elaborado por: Los Autores

Tabla 14 Historia de Usuario Gestión de Pulsos Cardíacos

HISTORIA DE USUARIO	
Número: 5	Usuario: Docentes, Alumnos
Nombre Historia: Gestión de Pulsos Cardíacos	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Medio
Puntos Estimados: 1	Interacción Asignada: 1
Programador Responsable: Sergio Palacios y Ángela Román	
Descripción: El docente tendrá la facilidad de establecer los diferentes pulsos cardíacos los mismos que están orientados a colapso obstétrico, los cuales son: Bradicardia (Alto), Normal, Taquicardia (Bajo) y Paro	
Observaciones: Si el docente o alumno a no presencia ningún cambio es necesario reiniciar el simulador debido a una pérdida de conexión, esto sucede a que existe inactividad entre la página y el simulador por un tiempo prolongado.	

Nota. Esta tabla muestra la historia de usuario al gestionar los pulsos cardíacos

Elaborado por: Los Autores

Tabla 15 Historia de Usuario Gestión de Labios

HISTORIA DE USUARIO	
Número: 5	Usuario: Docentes, Alumnos
Nombre Historia: Gestión de Pulsos Cardíacos	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Medio
Puntos Estimados: 1	Interacción Asignada: 1
Programador Responsable: Sergio Palacios y Ángela Román	
Descripción: El docente podrá cambiar el estado de los labios cuando la paciente entre en alguna crisis médica, los cuales se representarán mediante un cambio de color, tanto la Cianosis que se encenderá un led color morado y los labios normales que se representará con el led apagado.	
Observaciones: Si el docente o alumno no visualiza ningún cambio es necesario reiniciar el simulador debido a una pérdida de conexión, esto sucede a que existe inactividad entre la página y el simulador por un tiempo prolongado.	

Nota. Esta tabla muestra la historia de usuario al gestionar los pulsos cardíacos

Elaborado por: Los Autores

Tabla 16 Historia de Usuario Gestión de Bomba de Sangre

HISTORIA DE USUARIO	
Número: 5	Usuario: Docentes, Ingeniero a cargo, Alumnos
Nombre Historia: Gestión de Bomba de Sangre	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Medio
Puntos Estimados: 1	Interacción Asignada: 1
Programador Responsable: Sergio Palacios y Ángela Román	
Descripción: El docente tendrá la posibilidad de encender o apagar la bomba de sangre, esto dependerá si el alumno realiza una buena o mala práctica médica, la bomba contará con 2 estados, <u>empezar sangrado y detener sangrado</u> .	
Observaciones: Es necesario que el docente anticipe al ingeniero a cargo que verifique si cuenta con una bolsa de sangre y que la misma esté conectada a la bomba para evitar inconvenientes en la práctica.	
Si el docente o alumno no presencia ningún cambio a pesar de que la bomba esté conectada a una bolsa de sangre es necesario reiniciar el simulador debido a una pérdida de conexión, esto sucede a que existe inactividad entre la página y el simulador por un tiempo prolongado.	

Nota. Esta tabla muestra la historia de usuario al gestionar los cambios de la bomba de sangre

Elaborado por: Los Autores

4.3 Asignación de Roles de Usuario.

En este apartado se detallará la asignación de roles del presente proyecto tal como se indica en la Tabla 17.

Tabla 17 Asignación de roles de usuario

Nota. Esta tabla muestra las asignaciones de roles a cada uno de los encargados

Elaborado por: Los Autores

4.4 Plan de entrega del Proyecto

Es necesario elaborar un plan de entrega del proyecto que ayude a determinar las fechas de inicio y fin de cada historia de usuario a realizar, para esto se agrupara cada

ROLES	ASIGNADO A:
Programador	Sergio Palacios
Técnico de Telecomunicaciones	Ángela Román
Encargado de Pruebas (Tester)	Ing. David Erazo
Encargado de Seguimiento (Tracker)	Ing. Julio Proaño
Cliente	Ing. David Erazo
ROLES	ASIGNADO A:
Programador	Sergio Palacios
Técnico de Telecomunicaciones	Ángela Román
Encargado de Pruebas (Tester)	Ing. David Erazo
Encargado de Seguimiento (Tracker)	Ing. Julio Proaño
Cliente	Ing. David Erazo

historia de usuario con su respectiva iteración y la prioridad que ya se le asigno.

En la Tabla 18 se especificará el plan de entrega.

Tabla 18 Plan de entrega del Proyecto

Historia	Iteración	Prioridad	Fecha Inicio	Fecha Fin
1	1	Alta	01/07/2020	10/07/2020
2	1	Alta	13/07/2020	20/07/2020
3	2	Alta	23/07/2020	30/07/2020
4	2	Alta	03/08/2020	10/08/2020
5	3	Alta	13/08/2020	20/08/2020
6	3	Alta	24/08/2020	31/08/2020
7	3	Alta	01/09/2020	08/09/2020

Nota. Esta tabla muestra el plan de entrega del proyecto el cual especifica fechas tentativas

Elaborado por: Los Autores

4.5 Iteraciones

Una vez detalladas todas las historias de usuario, es necesario dividir las en etapas o iteraciones asignando un conjunto de historias por cada iteración, las cuales estarán identificadas por códigos facilitando el desarrollo del proyecto.

4.5.1 Primera Iteración

Para la primera iteración se utilizará las historias de acceso y gestión de credenciales multimedia las cuales se detallarán en la Tabla 19.

Tabla 19 Primera Iteración

CÓDIGO	HISTORIAS DE USUARIO
PI1	Acceso al Sistema
PI2	Gestión de Credenciales Multimedia

Nota. Esta tabla muestra la distribución de historias en la primera iteración

Elaborado por: Los Autores

4.5.1.1 Tareas de Ingeniería

En la Tabla 20 se indica todas las tareas que se van a realizar de acuerdo a la primera iteración ya mencionada, de igual manera en las tablas 21-25 se definirán de mejor manera cada una de las tareas.

Tabla 20 Tareas de Ingeniería

NÚMERO DE TAREA	CÓDIGO DE ITERACIÓN	NOMBRE DE LA TAREA
1	PI1	Diseño de la interfaz de acceso al sistema
2	PI1	Validación e Implementación de acceso
3	PI2	Adaptación de todos los sonidos del simulador
4	PI2	Admitir que el usuario cargue sonidos en SoundCloud
5	PI2	Tener acceso a las credenciales para gestionar los sonidos

Nota. Esta tabla muestra las tareas de ingeniería asociadas a la iteración 1

Elaborado por: Los Autores

4.5.1.1.1 Descripción de Tareas de Ingeniería

Tabla 21 Tarea de Ingeniería 1 para la Iteración 1

TAREA DE INGENIERIA	
Número de Tarea: 1	Código de Iteración: PI1
Nombre de la Tarea: Diseño de la interfaz de acceso al sistema	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 01/07/2020	Fecha Fin: 05/07/2020
Programador Responsable: Sergio Palacios	
Descripción: Se realiza el entorno gráfico para que sea amigable con el usuario.	

Nota. Esta tabla indica la descripción de la tarea de Ingeniería 1 para la Iteración 1

Elaborado por: Los Autores

Tabla 22 Tarea de Ingeniería 2 para la Iteración 1

TAREA DE INGENIERIA	
Número de Tarea: 2	Código de Iteración: PI1
Nombre de la Tarea: Validación e Implementación de acceso	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 06/07/2020	Fecha Fin: 10/07/2020
Programador Responsable: Ángela Román	
Descripción: Se necesita asignar y validar la IP por la cual el usuario va a tener acceso a la página.	

Nota. Esta tabla indica la descripción de la tarea de Ingeniería 2 para la Iteración 1

Elaborado por: Los Autores

Tabla 23 Tarea de Ingeniería 3 para la Iteración 1

TAREA DE INGENIERIA	
Número de Tarea: 3	Código de Iteración: PI2
Nombre de la Tarea: Adaptación de todos los sonidos del simulador en SoundCloud	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.2
Fecha Inicio: 13/07/2020	Fecha Fin: 14/07/2020
Programador Responsable: Sergio Palacios	
Descripción: Se debe crear una cuenta en SoundCloud permitiendo almacenar todos los sonidos del sistema.	

Nota. Esta tabla indica la descripción de la tarea de Ingeniería 3 para la Iteración 1

Elaborado por: Los Autores

Tabla 24 Tarea de Ingeniería 4 para la Iteración 1

TAREA DE INGENIERIA	
Número de Tarea: 4	Código de Iteración: PI2
Nombre de la Tarea: Admitir que el usuario cargue sonidos en SoundCloud	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.4
Fecha Inicio: 14/07/2020	Fecha Fin: 17/07/2020
Programador Responsable: Sergio Palacios	
Descripción: Se creará una opción dentro de un menú permitiendo que el docente acceda a la plataforma de sonidos.	

Nota. Esta tabla indica la descripción de la tarea de Ingeniería 4 para la Iteración 1

Elaborado por: Los Autores

Tabla 25 Tarea de Ingeniería 5 para la Iteración 1

TAREA DE INGENIERIA	
Número de Tarea: 5	Código de Iteración: PI2
Nombre de la Tarea: Tener acceso a las credenciales para gestionar los sonidos	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.4
Fecha Inicio: 17/07/2020	Fecha Fin: 20/07/2020
Programador Responsable: Sergio Palacios	
Descripción: Se creará una opción donde el docente pueda descargar las credenciales correspondientes para ingresar a la plataforma de sonidos.	

Nota. Esta tabla indica la descripción de la tarea de Ingeniería 5 para la Iteración 1

Elaborado por: Los Autores

4.5.2 Segunda Iteración

Para la segunda iteración se utilizarán las historias de gestión de ojos y labios, las cuales se detallarán en la Tabla 26.

Tabla 26 Segunda Iteración

CÓDIGO	HISTORIAS DE USUARIO
SI1	Gestión de Movimientos de ojos
SI2	Gestión de Labios

Nota. Esta tabla muestra la distribución de historias en la segunda iteración

Elaborado por: Los Autores

4.5.2.1 Tareas de Ingeniería

En la Tabla 27 se indica todas las tareas que se van a realizar de acuerdo a la segunda iteración ya mencionada, de igual manera en las tablas 28-33 se definirán de mejor manera cada una de las tareas.

Tabla 27 Tareas de Ingeniería

NÚMERO DE TAREA	CÓDIGO DE ITERACIÓN	NOMBRE DE LA TAREA
1	SI1	Diseño de botones para las diferentes interacciones con los ojos del simulador
2	SI1	Validación y manejo de las variables enviadas por cada uno de los movimientos de los ojos.
3	SI1	Verificación de cada uno de los movimientos en el simulador
4	SI2	Diseño de botones que permitan interactuar con los labios del simulador
5	SI2	Validación y manejo de las variables enviadas por cada una de las interacciones de los labios.
6	SI2	Verificación de cada una de las interacciones en el simulador

Nota. Esta tabla muestra las tareas de ingeniería asociadas a la iteración 2

Elaborado por: Los Autores

4.5.2.1.1 Descripción de Tareas de Ingeniería

Tabla 28 Tarea de Ingeniería 1 para la Iteración 2

TAREA DE INGENIERIA	
Número de Tarea: 1	Código de Iteración: SI1
Nombre de la Tarea: Diseño de botones para las diferentes interacciones con los ojos del simulador	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.4
Fecha Inicio: 23/07/2020	Fecha Fin: 26/07/2020
Programador Responsable: Sergio Palacios	
Descripción: Se realiza la creación y distribución de los botones que hacen referencia a cada uno de los movimientos de los ojos del simulador.	

Nota. Esta tabla indica la descripción de la tarea de Ingeniería 1 para la Iteración 2

Elaborado por: Los Autores

Tabla 29 Tarea de Ingeniería 2 para la Iteración 1

TAREA DE INGENIERIA	
Número de Tarea: 2	Código de Iteración: SI1
Nombre de la Tarea: Validación y manejo de las variables enviadas por cada uno de los movimientos de los ojos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.4
Fecha Inicio: 26/07/2020	Fecha Fin: 29/07/2020
Programador Responsable: Ángela Román	
Descripción: Se necesita identificar cada una de las variables recibidas por la página y asignarlas a las acciones correspondientes.	

Nota. Esta tabla indica la descripción de la tarea de Ingeniería 2 para la Iteración 2

Elaborado por: Los Autores

Tabla 30 Tarea de Ingeniería 3 para la Iteración 1

TAREA DE INGENIERIA	
Número de Tarea: 3	Código de Iteración: SI1
Nombre de la Tarea: Verificación de cada uno de los movimientos en el simulador	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.2
Fecha Inicio: 29/07/2020	Fecha Fin: 30/07/2020
Programador Responsable: Ángela Román	
Descripción: Se debe verificar cada uno de los movimientos de los ojos, comprobando que la asignación de variables haya sido correcta.	

Nota. Esta tabla indica la descripción de la tarea de Ingeniería 3 para la Iteración 2

Elaborado por: Los Autores

Tabla 31 Tarea de Ingeniería 4 para la Iteración 2

TAREA DE INGENIERIA	
Número de Tarea: 4	Código de Iteración: SI2
Nombre de la Tarea: Diseño de botones que permitan interactuar con los labios del simulador	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.4
Fecha Inicio: 03/08/2020	Fecha Fin: 06/08/2020
Programador Responsable: Sergio Palacios	
Descripción: Se realiza la creación y distribución de los botones que hacen referencia a los estados de los labios en el simulador.	

Nota. Esta tabla indica la descripción de la tarea de Ingeniería 4 para la Iteración 2

Elaborado por: Los Autores

Tabla 32 Tarea de Ingeniería 5 para la Iteración 2

TAREA DE INGENIERIA	
Número de Tarea: 5	Código de Iteración: SI2
Nombre de la Tarea: Validación y manejo de las variables enviadas por cada una de las interacciones de los labios.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.4
Fecha Inicio: 06/08/2020	Fecha Fin: 09/08/2020
Programador Responsable: Ángela Román	
Descripción: Se necesita identificar cada una de las variables recibidas por la página y asignarlas a las acciones correspondiente.	

Nota. Esta tabla indica la descripción de la tarea de Ingeniería 5 para la Iteración 2

Elaborado por: Los Autores

Tabla 33 Tarea de Ingeniería 6 para la Iteración 2

TAREA DE INGENIERIA	
Número de Tarea: 6	Código de Iteración: SI2
Nombre de la Tarea: Verificación de cada una de las interacciones en el simulador	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.2
Fecha Inicio: 09/08/2020	Fecha Fin: 10/08/2020
Programador Responsable: Ángela Román	
Descripción: Se debe verificar cada una de las interacciones de los labios, comprobando que la asignación de variables haya sido correcta.	

Nota. Esta tabla indica la descripción de la tarea de Ingeniería 6 para la Iteración 2

Elaborado por: Los Autores

4.5.3 Tercera Iteración

Para la segunda iteración se utilizarán las historias de gestión de ojos y labios, las cuales se detallarán en la Tabla 34.

Tabla 34 Tercera Iteración

CÓDIGO	HISTORIAS DE USUARIO
TI1	Gestión de pulsos cardiacos
TI2	Gestión de sonidos torácicos
TI3	Gestión de bomba de sangre

Nota. Esta tabla muestra la distribución de historias en la tercera iteración

Elaborado por: Los Autores

4.5.3.1 Tareas de Ingeniería

En la Tabla 35 se indica todas las tareas que se van a realizar de acuerdo a la tercera iteración ya mencionada, de igual manera en las tablas 36-44 se definirán de mejor manera cada una de las tareas.

Tabla 35 Tareas de Ingeniería

NÚMERO DE TAREA	CÓDIGO DE ITERACIÓN	NOMBRE DE LA TAREA
1	TI1	Diseño de botones que interactúen con los pulsos cardiacos del simulador
2	TI1	Validación y manejo de cada una de las interacciones de los pulsos cardiacos enviadas por la página.
3	TI1	Verificación de cada uno de los pulsos cardiacos en el simulador
4	TI2	Diseño de botones que interactúen con los sonidos torácicos del simulador
5	TI2	Validación y manejo de cada una de las interacciones de los sonidos torácicos enviados por la página.
6	TI2	Verificación de cada uno de los sonidos torácicos en el simulador
7	TI3	Diseño de botones que interactúen con la bomba de sangre del simulador
8	TI3	Validación y manejo de cada uno de los estados de la bomba de sangre enviadas por la página.
9	TI3	Verificación de cada uno de los estados en el simulador

Nota. Esta tabla muestra las tareas de ingeniería asociadas a la iteración 3

Elaborado por: Los Autores

4.5.3.1.1 Descripción de Tareas de Ingeniería

Tabla 36 Tarea de Ingeniería 1 para la Iteración 3

TAREA DE INGENIERIA	
Número de Tarea: 1	Código de Iteración: TI1
Nombre de la Tarea: Diseño de botones que interactúen con los pulsos cardiacos del simulador	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.4
Fecha Inicio: 13/08/2020	Fecha Fin: 16/08/2020

Programador Responsable: Sergio Palacios
Descripción: Se realiza la creación y distribución de los botones que hacen referencia a cada uno de los pulsos cardiacos del simulador.

Nota. Esta tabla indica la descripción de la tarea de Ingeniería 1 para la Iteración 2

Elaborado por: Los Autores

Tabla 37 Tarea de Ingeniería 2 para la Iteración 3

TAREA DE INGENIERIA	
Número de Tarea: 2	Código de Iteración: TI1
Nombre de la Tarea: Validación y manejo de cada una de las interacciones de los pulsos cardiacos enviadas por la página.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.4
Fecha Inicio: 16/08/2020	Fecha Fin: 19/08/2020
Programador Responsable: Ángela Román	
Descripción: Se necesita identificar cada una de las variables recibidas por la página y asignarlas a las acciones correspondientes.	

Nota. Esta tabla indica la descripción de la tarea de Ingeniería 2 para la Iteración 2

Elaborado por: Los Autores

Tabla 38 Tarea de Ingeniería 3 para la Iteración 3

TAREA DE INGENIERIA	
Número de Tarea: 3	Código de Iteración: TI1
Nombre de la Tarea: Verificación de cada uno de los pulsos cardiacos en el simulador	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.2
Fecha Inicio: 19/08/2020	Fecha Fin: 20/08/2020
Programador Responsable: Ángela Román	
Descripción: Se debe validar cada uno de los pulsos cardiacos, verificando que la asignación de variables haya sido correcta.	

Nota. Esta tabla indica la descripción de la tarea de Ingeniería 3 para la Iteración 3

Elaborado por: Los Autores

Tabla 39 Tarea de Ingeniería 4 para la Iteración 3

TAREA DE INGENIERIA	
Número de Tarea: 4	Código de Iteración: TI2
Nombre de la Tarea: Diseño de botones que interactúen con los sonidos torácicos del simulador	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.4
Fecha Inicio: 24/08/2020	Fecha Fin: 27/08/2020
Programador Responsable: Sergio Palacios	
Descripción: Se realiza la creación y distribución de los botones que hacen referencia a cada uno de los sonidos torácicos del simulador.	

Nota. Esta tabla indica la descripción de la tarea de Ingeniería 4 para la Iteración 3

Elaborado por: Los Autores

Tabla 40 Tarea de Ingeniería 5 para la Iteración 3

TAREA DE INGENIERIA	
Número de Tarea: 5	Código de Iteración: TI2
Nombre de la Tarea: Validación y manejo de cada una de las interacciones de los sonidos torácicos enviados por la página.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.4
Fecha Inicio: 27/08/2020	Fecha Fin: 30/08/2020
Programador Responsable: Ángela Román	
Descripción: Se necesita identificar cada una de las variables recibidas por la página y asignarlas a las acciones correspondiente.	

Nota. Esta tabla indica la descripción de la tarea de Ingeniería 5 para la Iteración 3

Elaborado por: Los Autores

Tabla 41 Tarea de Ingeniería 6 para la Iteración 3

TAREA DE INGENIERIA	
Número de Tarea: 6	Código de Iteración: TI2
Nombre de la Tarea: Verificación de cada uno de los sonidos torácicos en el simulador	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.2
Fecha Inicio: 30/08/2020	Fecha Fin: 31/08/2020
Programador Responsable: Ángela Román	
Descripción: Se debe validar cada uno de los sonidos torácicos, verificando que la asignación de variables haya sido correcta.	

Nota. Esta tabla indica la descripción de la tarea de Ingeniería 6 para la Iteración 3

Elaborado por: Los Autores

Tabla 42 Tarea de Ingeniería 7 para la Iteración 3

TAREA DE INGENIERIA	
Número de Tarea: 7	Código de Iteración: TI3
Nombre de la Tarea: Diseño de botones que interactúen con la bomba de sangre del simulador	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.4
Fecha Inicio: 01/09/2020	Fecha Fin: 04/09/2020
Programador Responsable: Sergio Palacios	
Descripción: Se realiza la creación y distribución de los botones que hacen referencia a cada uno de los estados de la bomba de sangre del simulador.	

Nota. Esta tabla indica la descripción de la tarea de Ingeniería 7 para la Iteración 3

Elaborado por: Los Autores

Tabla 43 Tarea de Ingeniería 8 para la Iteración 3

TAREA DE INGENIERIA	
Número de Tarea: 8	Código de Iteración: TI3
Nombre de la Tarea: Validación y manejo de cada uno de los estados de la bomba de sangre enviadas por la página.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.4
Fecha Inicio: 04/09/2020	Fecha Fin: 07/09/2020
Programador Responsable: Ángela Román	
Descripción: Se necesita identificar cada una de las variables recibidas por la página y asignarlas a las acciones correspondiente.	

Nota. Esta tabla indica la descripción de la tarea de Ingeniería 8 para la Iteración 3

Elaborado por: Los Autores

Tabla 44 Tarea de Ingeniería 9 para la Iteración 3

TAREA DE INGENIERIA	
Número de Tarea: 9	Código de Iteración: TI3
Nombre de la Tarea: Verificación de cada uno de los estados en el simulador	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.2
Fecha Inicio: 07/09/2020	Fecha Fin: 08/09/2020
Programador Responsable: Ángela Román	
Descripción: Se debe validar cada uno de los estados de la bomba de sangre, verificando que la asignación de variables haya sido correcta.	

Nota. Esta tabla indica la descripción de la tarea de Ingeniería 9 para la Iteración 3

Elaborado por: Los Autores

CAPÍTULO V

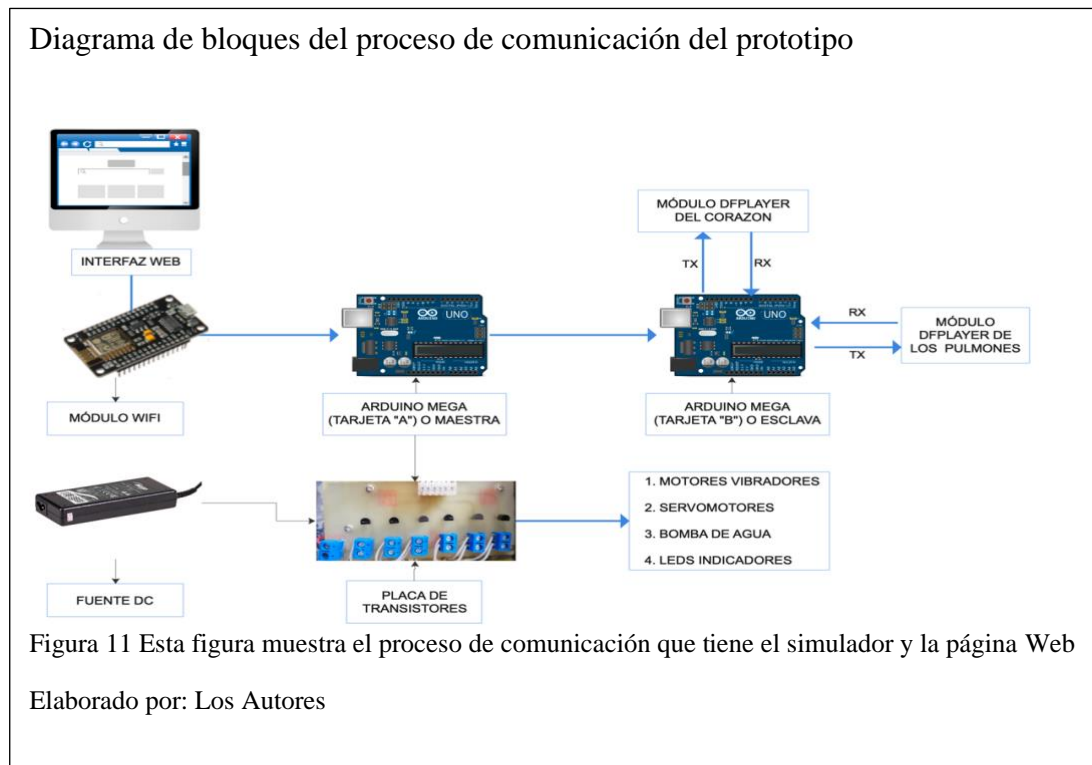
DISEÑO

En este capítulo se presenta la fase de diseño, en la que se elaboró un diagrama de bloques el cual permitió identificar el funcionamiento interno del simulador, así también se construyó los diagramas uml correspondientes al diseño de la página Web. Seguidamente se diseñaron las tarjetas CRC facilitando el análisis de clases y responsabilidades que tiene cada una de ellas, por último, se diseñó el bosquejo principal donde se englobo las responsabilidades anteriormente mencionadas en las tarjetas. Esta fase permitió adquirir una comprensión conjunta tanto para el desarrollador como para el usuario, brindando simplicidad al diseño de la página Web logrando seguir las recomendaciones dictadas en la metodología

5.1 Diagrama de bloques del proceso de comunicación del simulador

La Figura 11 representa el proceso de comunicación entre la página Web y el fantoma, de igual manera se detallan todos los componentes que posee el simulador y como estos se comunican entre si logrando un correcto flujo de información. Una vez diseñada la página Web esta se podrá visualizar desde cualquier navegador a través de una dirección IP, para que esto suceda es necesario configurar el módulo Wi-Fi de tal manera que pueda acceder a la red de la clínica de simulación. Dicho módulo tendrá una comunicación directa con la tarjeta maestra de Arduino permitiendo recibir todas las peticiones enviadas por el usuario, en esta placa se encuentran conectados todos los elementos actuadores del simulador como: motores vibradores, servomotores, bomba de sangre y leds, los cuales se encuentran conectadas a una placa de transistores alimentada por una fuente DC.

Por otra parte, la tarjeta maestra está conectada directamente a la tarjeta B o (esclava), en esta nueva tarjeta estarán conectados los moduladores de sonidos tanto para el corazón como para los pulmones, se la denomina esclava porque la función principal de esta placa es procesar y enviar la información recibida de la tarjeta maestra a cada uno de los elementos actuadores.



De igual manera se construyó algunos diagramas UML permitiendo modelar todas las funciones que tiene la página Web y las diferentes acciones que el docente posee sobre la misma.

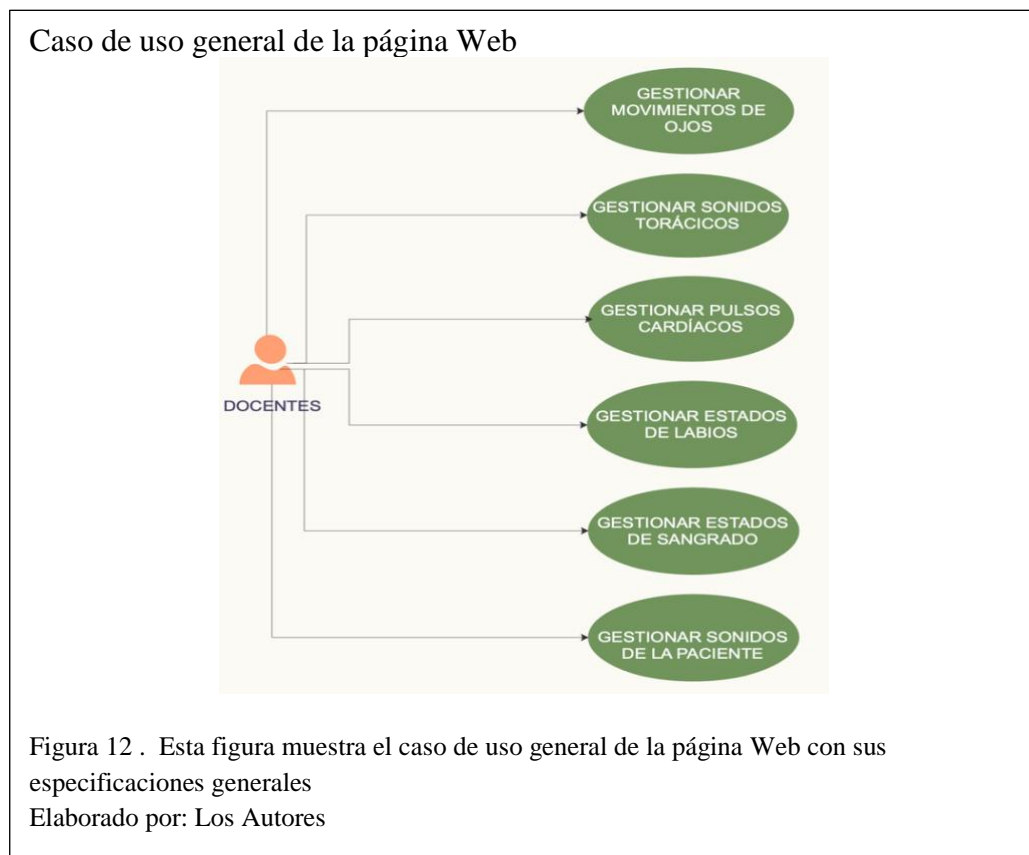
5.2 Diagramas UML

Estos diagramas están formados por varios elementos que permiten representar de manera gráfica las funciones del sistema a desarrollar, por lo que es necesario diseñar los diagramas de casos de uso y de estado, con el objetivo de plasmar todas las

acciones que el usuario podrá realizar, así como los diferentes estados que tendrá la página Web al momento de recibir una petición.

5.2.1 Diagramas de casos de uso.

La Figura 12 muestra el caso de uso general, en el cual se encuentran todas las especificaciones que tendrá el docente sobre el sistema, de igual manera en las figuras 13-18 se detallan cada una de estas.



La Figura 13 muestra el caso de uso específico el cual detalla las acciones que engloba el caso de uso referente a los movimientos de ojos del simulador.

Caso de uso gestionar movimientos de ojos

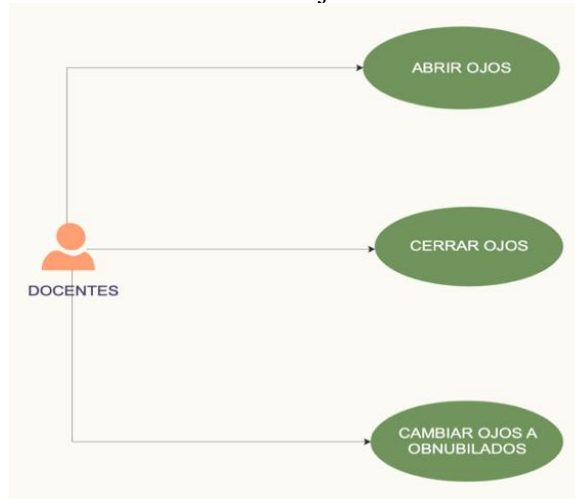


Figura 13 Esta figura muestra el caso de uso específico referente al movimiento de ojos
Elaborado por: Los Autores

La Figura 14 indica el caso de uso específico el cual detalla las acciones que engloba el caso de uso referente a los sonidos torácicos del simulador.

Caso de uso gestionar sonidos torácicos

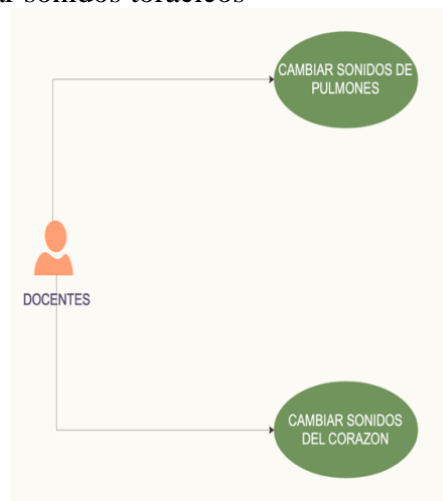


Figura 14 Esta figura muestra el caso de uso específico referente a los sonidos del tórax
Elaborado por: Los Autores

En la Figura 15 se observa el caso de uso específico el cual detalla las acciones que engloba el caso de uso referente a los pulsos cardíacos del simulador.

Caso de uso gestionar pulsos cardíacos

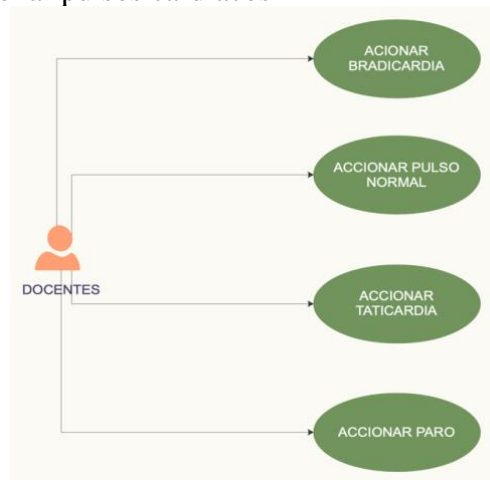


Figura 15 Esta figura muestra el caso de uso específico referente a los pulsos cardíacos. Elaborado por: Los Autores

En la Figura 16 se muestra el caso de uso específico el cual detalla las acciones que engloba el caso de uso referente a los estados de los labios en el simulador.

Caso de uso gestionar estados de labios

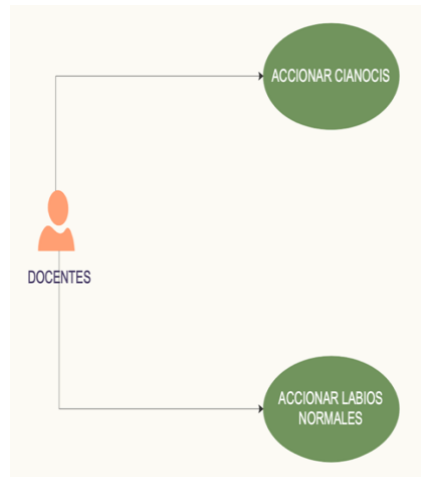


Figura 16 Esta figura muestra el caso de uso específico referente a los estados de labios. Elaborado por: Los Autores

En la Figura 17 se indica el caso de uso específico el cual detalla las acciones que engloba el caso de uso referente al sangrado que tendrá el simulador.

Gestionar estados de sangrado



Figura 17 Esta figura muestra el caso de uso específico referente a los estados del sangrado.

Elaborado por: Los Autores

En la Figura 18 se indica el caso de uso específico el cual detalla las acciones que engloba el caso de uso referente a los sonidos que tendrá la paciente.

Caso de uso gestionar sonidos de la paciente

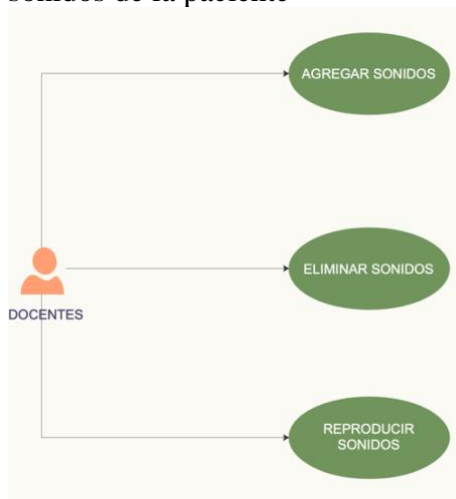


Figura 18 Esta figura muestra el caso de uso específico referente a los sonidos de la paciente.

Elaborado por: Los Autores

5.2.2 Diagrama de estado

En la figura 19 se presentará el diagrama de estado de la página, con el objeto de modelar el ciclo de vida que esta tendrá ante la presencia de alguna condición o comportamiento enviada por el usuario.

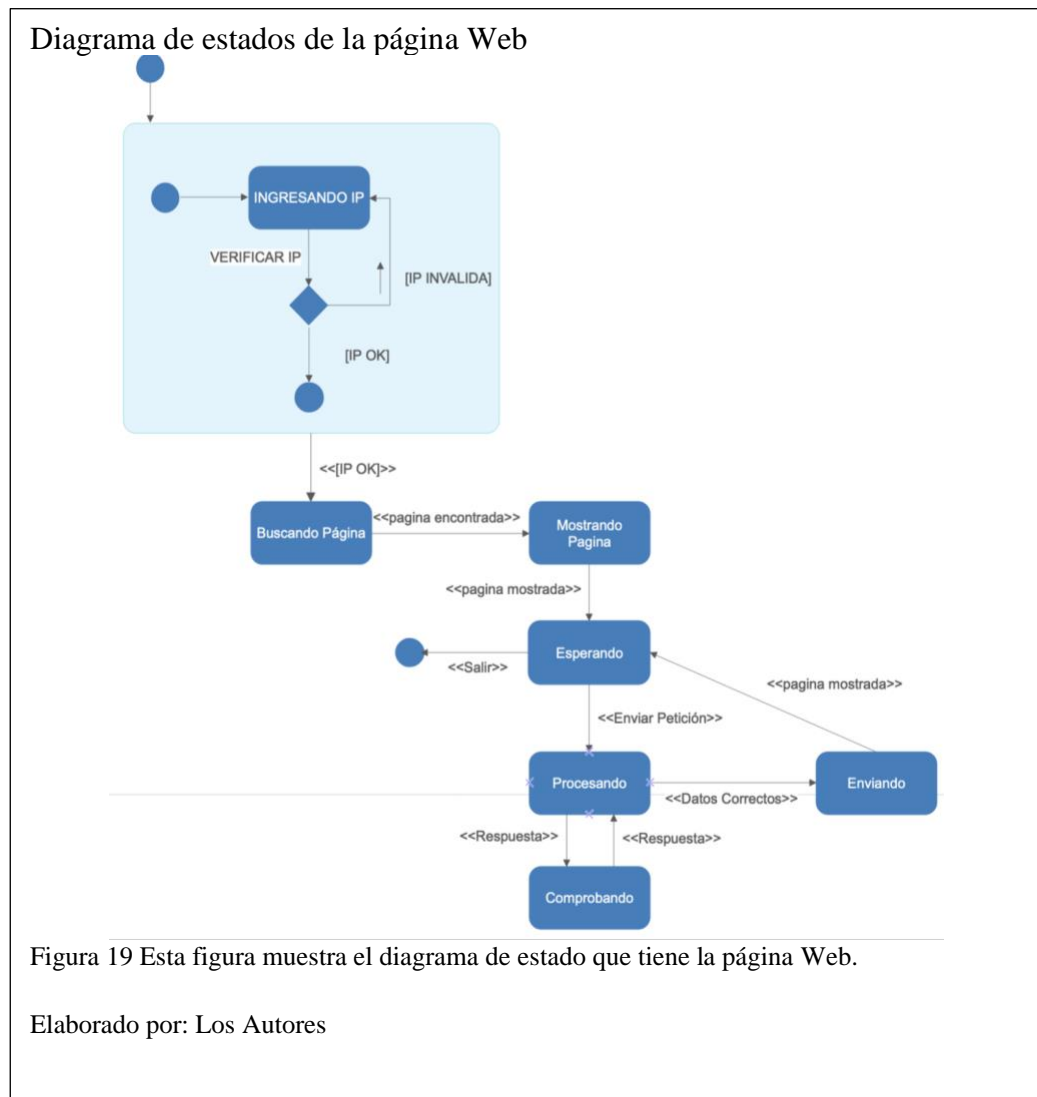


Figura 19 Esta figura muestra el diagrama de estado que tiene la página Web.

Elaborado por: Los Autores

En primera instancia, el ciclo de vida comienza cuando el usuario ingresa la IP en el navegador, la cual es verificada y en caso de ser incorrecta retorna a su estado inicial sin mostrar ningún mensaje al usuario, caso contrario da comienzo a un nuevo subproceso tal como se muestra en la Figura 20.

Ciclo inicial del sistema

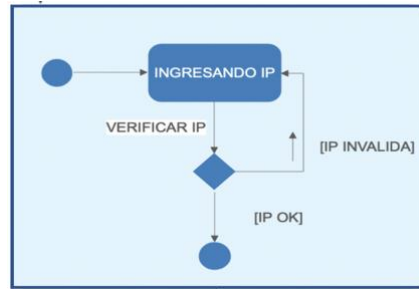


Figura 20 Esta figura indica el ciclo inicial de ingreso y validación de IP.

Elaborado por: Los Autores

Posteriormente, una vez validada la IP el sistema inicia un nuevo subproceso, el cual se encarga de buscar y mostrar la página Web alojada en el módulo Wi-Fi, la misma que se queda en espera de recibir alguna petición enviada por el usuario. Una vez recibida la petición el dato es procesado, comprobado y enviado al simulador. Terminada la petición la página se auto refresca y está lista para un nuevo requerimiento, tal como se indica en la Figura 21.

Ciclo de requerimientos de usuario

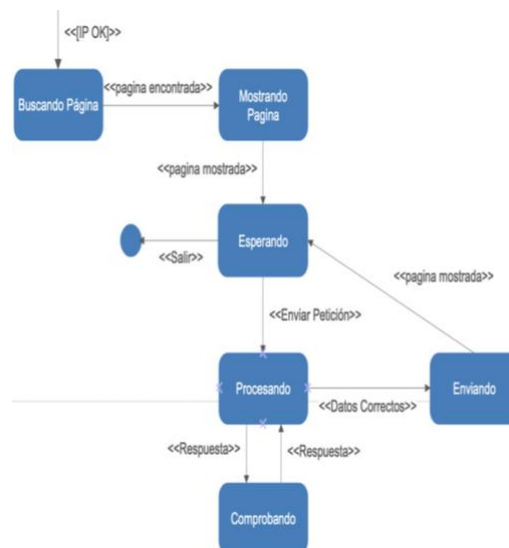


Figura 21 Esta figura indica el ciclo al momento que el usuario envía una petición.

Elaborado por: Los Autores

Luego que se hayan modelado todas las funciones de la página es necesario plasmarlas en tarjetas CRC, esto permitirá facilitar el proceso de diseño de una manera

simple y óptima, logrando de esta manera adquirir la capacidad de ver los problemas de una forma sencilla y sin tener que pensar tanto en el código, esto significa que basta pensar en la solución de una sola tarea y la misma pueda ser replicada en las demás.

5.3 Tarjeta CRC (Clase – Responsabilidades - Colaboradores)

En esta sección haremos uso de las tarjetas CRC que harán referencia a las responsabilidades que tiene nuestra clase principal en sus tres iteraciones anteriormente mencionadas, tal como se muestra en la tabla 45.

Tabla 45 Tarjeta CRC Control Fantoma

Control Fantoma	
Colaboración	Responsabilidades
Controlador Fantoma	Agregar sonidos de la paciente
	Reproducir sonidos de la paciente
	Abrir ojos
	Cerrar ojos
	Ojos obnubilados
	Reproducir sonidos del corazón
	Reproducir sonidos de los pulmones
	Cambiar pulsos cardiacos
	Modificar el estado de los labios
	Empezar hemorragia
	Detener hemorragia

Nota. Esta tabla muestra todas las responsabilidades de la clase principal.

Elaborado por: Los Autores

A continuación, se realizará el bosquejo de la página Web teniendo en cuenta las responsabilidades ya mencionadas.

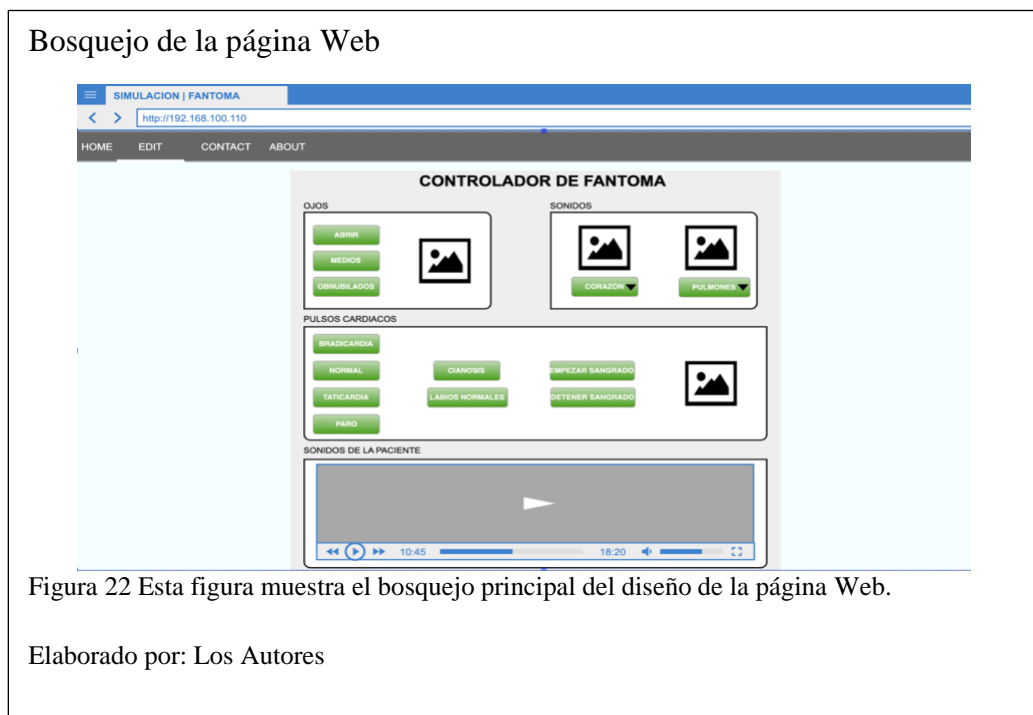
5.4 Bosquejo del Diseño

En este apartado se desarrollará un boceto que ayude en la planificación del diseño de la página Web tomando en cuenta todas las responsabilidades detalladas en la Tabla 46, posteriormente será revisado por el encargado de la clínica determinando si el bosquejo final es amigable con el usuario.

Se presentará un boceto final de la página Web utilizando la herramienta Pencil Project, esta herramienta cuenta con los elementos necesarios para obtener un diseño practico y amigable.

5.4.1 Bosquejo de la Página Web

En la Figura 22 se observa el bosquejo principal de la página Web la cual está compuesta por los siguientes elementos: barra de menú, contenedor de ojos, contenedor de sonidos y contenedor de pulsos, los mismos que se detallan en el siguiente capítulo.



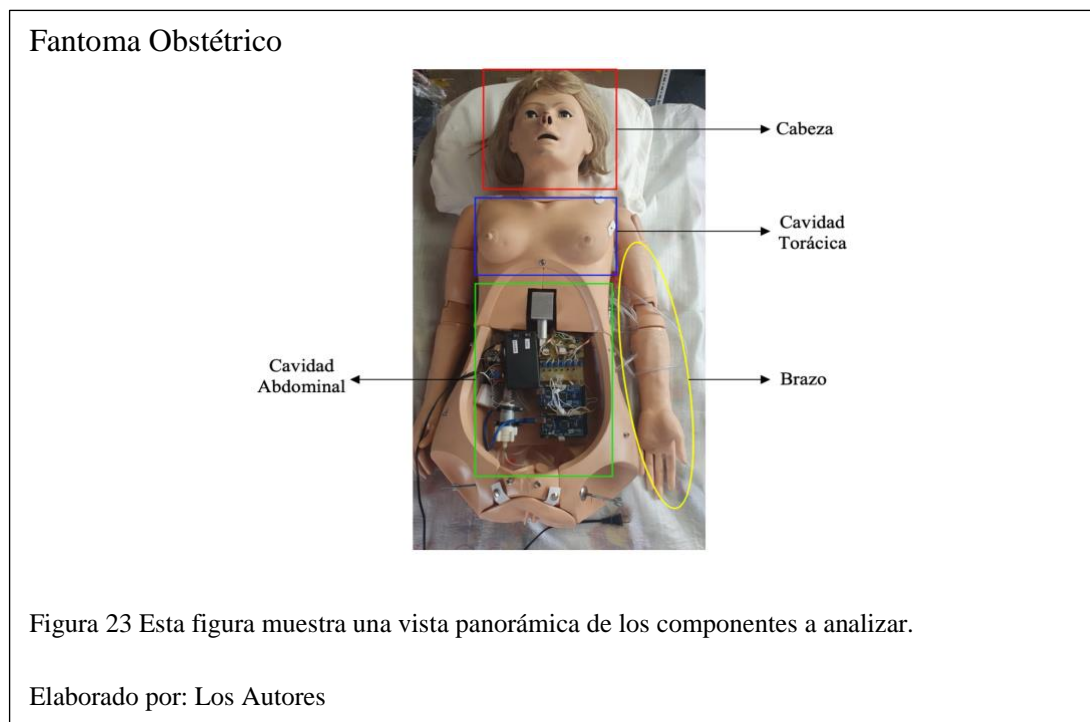
CAPÍTULO VI

DESARROLLO

En este capítulo se presenta un análisis inicial del fantoma tanto en sus condiciones como estructuración, esto permitió entender el proceso de comunicación que se está llevando a cabo, obteniendo bases sólidas para iniciar la solución. Asimismo, se realizó el módulo de comunicación, las interfaces del sistema y se analizó las herramientas utilizadas.

6.1 Análisis de estado inicial del Fantoma.

Para iniciar el desarrollo de la solución es necesario identificar los diferentes actuadores que se encuentran funcionando en el simulador, permitiendo comprender la conexión que existe entre cada uno de ellos y como estos se comunican a las placas de Arduino. Para esto es necesario analizar cada una de las partes que conforman el simulador como: cabeza, brazos, cavidad torácica y cavidad abdominal tal como muestra la Figura 23, las cuales se irán detallando de mejor manera a continuación.

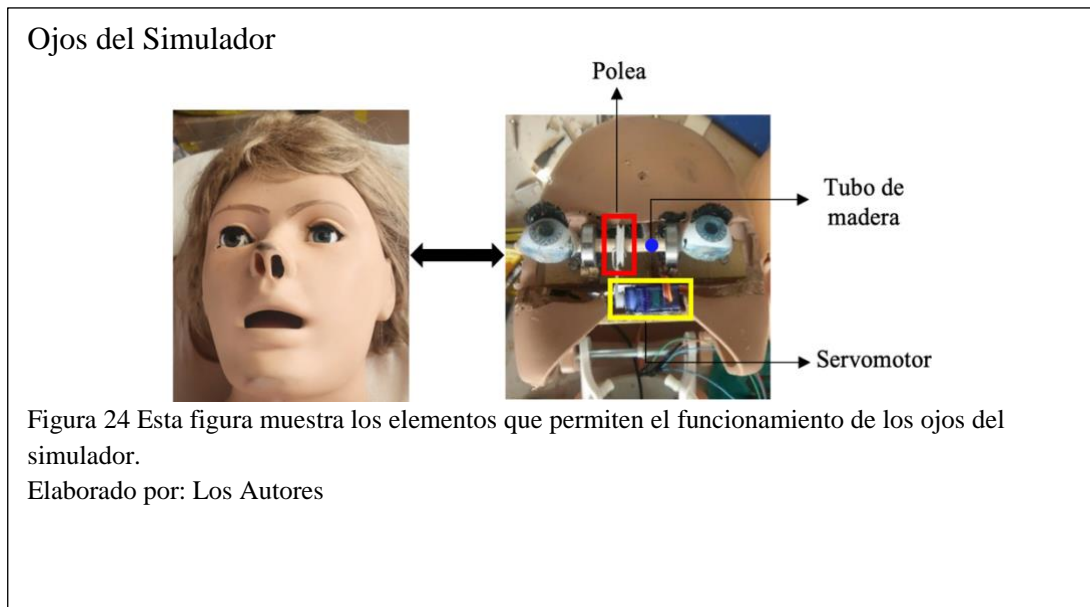


6.1.1 Análisis de la cabeza del simulador

Una vez que se hayan reconocido todas las partes del simulador es preciso desmontar la cabeza, con el objetivo de identificar los diferentes componentes que tienen tanto en los ojos como en la boca, los cuales se detallan a continuación.

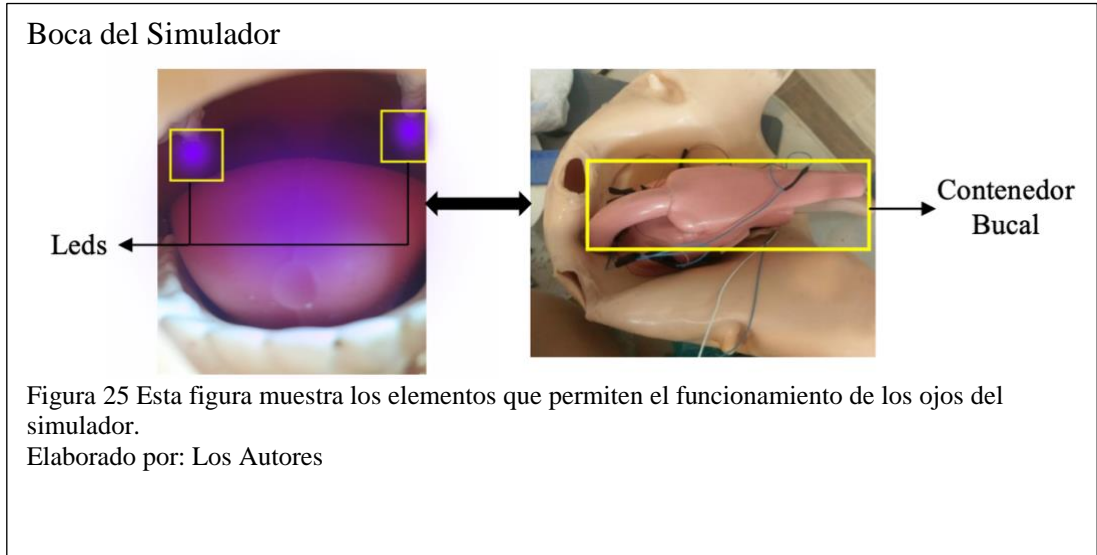
6.1.1.1 Ojos

Tal como se observa en la Figura 24 existe una serie de elementos que interactúan entre sí, permitiendo el correcto funcionamiento de los ojos. Este diseño se basa en una base de madera que permite sostener un tubo, al cual están anclados los dos ojos y en medio una polea que se encuentra conectada al servomotor por medio de una banda elástica.



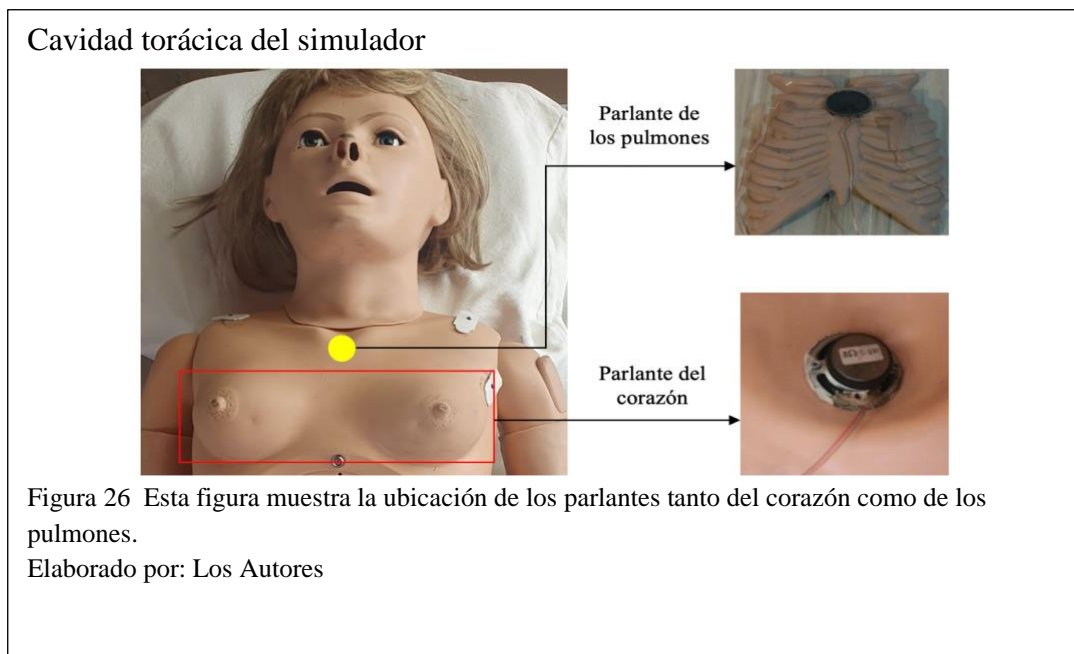
6.1.1.2 Boca

En la Figura 25 se observa dos leds de color morado conectados en paralelo, los cuales permiten emular el estado de cianosis de una paciente. Tanto la lengua, los leds y los dientes se encuentran alojados en un contenedor bucal el mismo que está sujeto a la piel de silicona del simulador.



6.1.2 Análisis de la cavidad torácica del simulador.

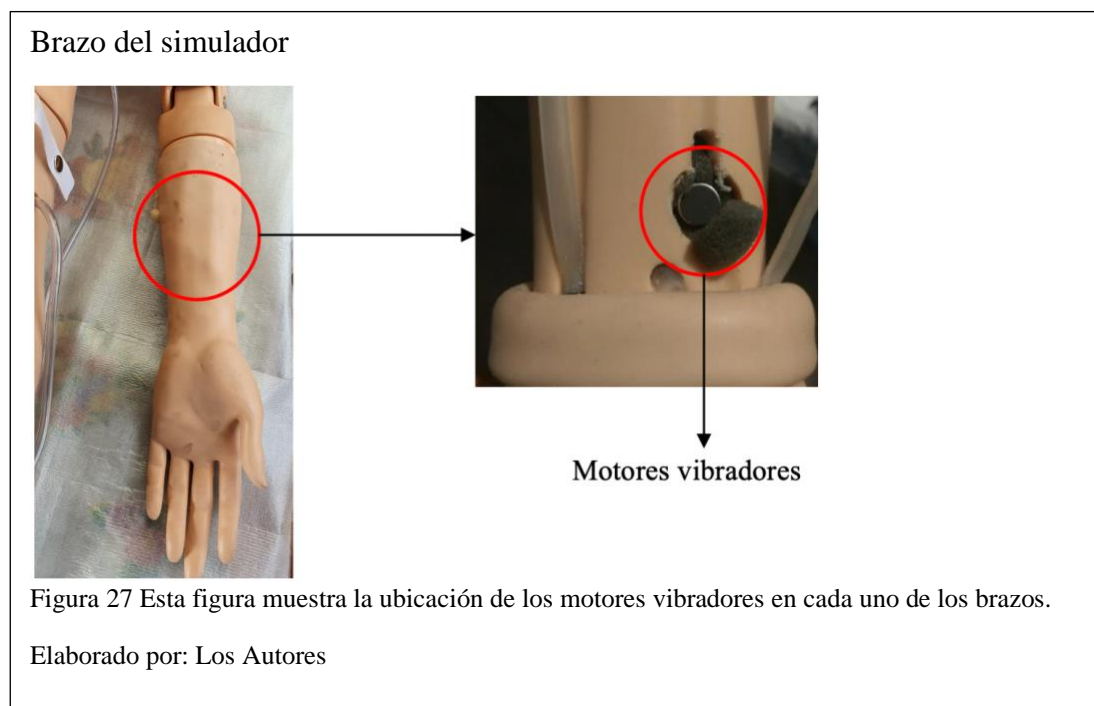
Por otra parte, la cavidad torácica se encuentra compuesta por 2 parlantes, el primer parlante hace referencia a los sonidos pulmonares de la paciente, el mismo que se encuentra sujeto a los pulmones del simulador. El segundo parlante corresponde a los sonidos del corazón tal como indica la Figura 26.



6.1.3 Análisis de los brazos del simulador.

Una vez desmontados ambos brazos se identificó que están compuestos por motores vibradores los cuales están ubicados en el antebrazo, tal como muestra la Figura 27.

Estos motores permiten simular los diferentes pulsos cardíacos que la paciente pueda presentar al momento de una complicación clínica.



6.1.4 Análisis de la cavidad abdominal del simulador.

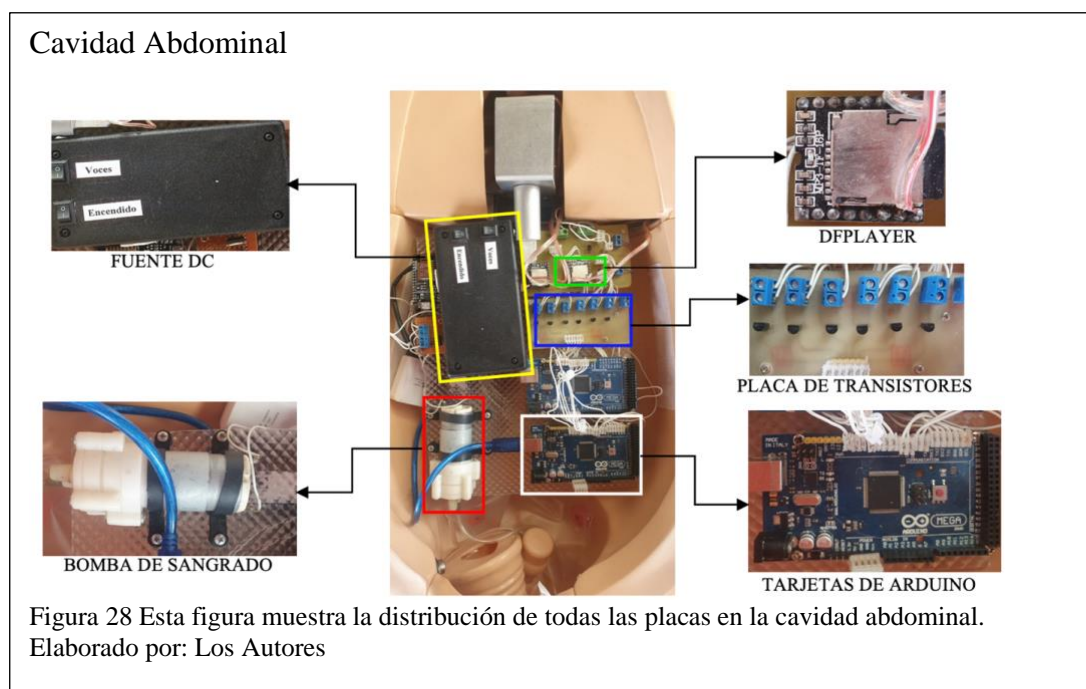
En este apartado se analizará todos los elementos que contiene la cavidad abdominal como: tarjetas de Arduino, bomba de sangre, fuente DC, DFPlayers y placa de transistores, tal como se observa en la Figura 28.

En primera instancia, la fuente DC cuenta con un cable de alimentación el cual es conectado de forma directa a 110 Vac, permitiendo el flujo de corriente a las diferentes placas controladoras.

Posteriormente, se observa la bomba de sangre que se encuentra conectada de forma directa a la tarjeta principal o maestra. Esta bomba es encargada de succionar toda la sangre alojada en una bolsa externa y expulsarla a través de una válvula, brindando un mejor realismo al realizar una simulación.

De igual manera, se evidencia una placa de transistores conectadas a las placas de Arduino, con el objetivo de amplificar la corriente recibida y enviarla de forma gradual a las diferentes partes del simulador ya mencionadas, esto evita que exista una caída de potencia y fallos en cualquier actuador.

Por último, se identificó 2 DFPlayers los cuales tienen una micro SD de 2GB cada una, estos almacenan todos los audios referentes a las complicaciones que puede tener el simulador.



Una vez analizado todos los componentes del simulador es necesario revisar la programación de cada una de las tarjetas, con el objetivo de identificar que parámetros son enviados a cada uno de los actuadores, lo cual se detallará a continuación.

6.2 Análisis de las tarjetas Arduino

Tal como se observa en la Figura 29 es necesario identificar todos los parámetros asignados a cada uno de los elementos del simulador, por lo que es necesario revisar la programación de cada una de las placas, esto será de ayuda al momento de programar la página Web y el módulo Wi-Fi.

Parámetros Tarjeta Maestra

```

56 //Asignación de instrucción para ojos      228 //Asignación de instrucción para sonidos de los pulmones      162 //Asignación de instrucción para sonidos del corazón
57 if (FI== 'a'){                             229 if(FI=='p'){                             163 if(FI=='h'){
58   v1='a';                                   230 v4 = 'p';                               164 v3 = 'h';
59 }                                           231 }                                         165 }
60 if (FI== 'b'){                             232 if(FI=='q'){                             166 if(FI=='i'){
61   v1='b';                                   233 v4 = 'q';                               167 v3 = 'i';
62 }                                           234 }                                         168 }
63 if (FI== 'c'){                             235 if(FI=='r'){                             169 if(FI=='j'){
64   v1='c';                                   236 v4 = 'r';                               170 v3 = 'j';
65 }                                           237 }                                         171 }
66 //Activación de ojos                      238 if(FI=='s'){                             172 if(FI=='k'){
67 if (v1=='a') //Abrir                       239 v4 = 's';                               173 v3 = 'k';
68 {                                           240 }                                         174 }
69 ojed.write(90);                            241 if(FI=='t'){                             175 if(FI=='l'){
70 }                                           242 v4 = 't';                               176 v3 = 'l';
71 if (v1== 'b') //Cerrar                    243 }                                         177 }
72 {                                           244 if(FI=='u'){                             178 if(FI=='m'){
73   ojed.write(150);                          245 v4 = 'u';                               179 v3 = 'm';
74 }                                           246 }                                         180 }
75 if (v1== 'c') //Medio abiertos            247 if(FI=='v'){                             181 if(FI=='n'){
76 {                                           248 v4 = 'v';                               182 v3 = 'n';
77   ojed.write(110);                          249 }                                         183 }
78 }                                           250 if(v4=='p'){                             184 if(FI=='o'){
                                                    295 //Asignación de variable para activación de bomba
285 //Asignación para activación de labios morados
286 if (FI== 'w'){                             300 if (FI== 'x'){
287   v5='w';                                   301   v6='x';
288 }                                           302 }
289 if (FI== 'y'){                             303 if (FI== 'u'){
290   v5='y';                                   304   v6='u';
291 }                                           305 }

```

Figura 29 Esta figura muestra todos los parámetros enviados a cada uno de los actuadores del simulador.

Fuente (Heredia Iza & Montaluisa Montaluisa, 2020)

Realizado este análisis es necesario organizarlo en una tabla, de tal manera que los parámetros sean fáciles de identificar al realizar la respectiva asignación en el módulo Wi-Fi, en consecuencia, estos parámetros serán detallados en la Tabla 47.

Tabla 46 Parámetros del simulador

OJOS	
DESCRIPCIÓN	PARÁMETRO ENVIADO
Abiertos	“b”
Cerrados	“a”
Obnubilados	“c”
SONIDOS PULMONARES	
Normal	“p”
Sibilancias (Espiratoria)	“q”
Estridor (Inspiratorio)	“r”
Crepitantes (Inspiratorio)	“s”
Subcrepitantes (Inspiratorio)	“t”
Estertores Húmedos (Inspiratorio)	“u”
Sin Sonido	“v”
SONIDOS DEL CORAZON	
Normal	“h”
Taquicardia	“i”
Soplo de Austin Flit	“j”
Soplo Diastólico	“k”
Soplo Holo Sistólico	“l”
Comunicación Interauricular	“m”
Soplo Septal	“n”
Sin Sonido	“o”
ESTADO DE LOS LABIOS	
Cianosis	“w”
Labios Normales	“y”

PULSOS CARDÍACOS	
Bradicardia	“d”
Normal	“e”
Taquicardia	“f”
Paro	“g”
BOMBA DE SANGRE	
Empezar Sangrado	“x”
Detener Sangrado	“z”

Nota. Esta tabla muestra todos los parámetros que permiten controlar el simulador, por (Palacios Sergio, Román Ángela, 2020).

Una vez identificados todos los parámetros que controlan al simulador, es necesario configurar el medio de comunicación por el cual se transmitirá la información, seguidamente detallados.

6.3 Configuración del módulo de comunicación

En primera instancia, es necesario contar con el software Arduino IDE, la cual permitirá desarrollar y validar la programación del módulo ESP8266. Para implementar la comunicación de dicho módulo es necesario definir el nombre y la contraseña de la red de la clínica de simulación, si los parámetros ingresados son correctos se configurará una IP estática que pertenezca a la red, la cual se convertirá en el identificador del simulador. Realizada toda la configuración es necesario verificar que todo este correcto, por lo cual se establecerá el puerto serial 9600 permitiendo validar la conexión como se observa en la Figura 30.

Configuración del módulo de comunicación

.....

WiFi Conectado....

Servidor inicializado

Use this URL to connect: http://192.168.100.150/

```
#include <ESP8266WiFi.h> // Incluir librería del módulo usado
#include <SoftwareSerial.h>

const char* ssid = "Sergio_2.4G"; // Definir el nombre de la red
const char* password = "1721326559.SERGIO.1994"; // Definir la contraseña de la red

SoftwareSerial s(3,1); // Inicializar puertos de comunicación RX|TX

IPAddress staticIP(192,168,100,150); // Declarar una IP estática dentro de la red
IPAddress gateway(192,168,100,1); // Declarar la puerta de enlace
IPAddress subnet(255,255,255,0); // Declarar la máscara de red

WiFiServer server(80); // Definir el puerto del servidor HTTP

void setup(void) {
  Serial.begin(9600); // Iniciar el puerto de escritura 9600
  Serial.print("Conectando a.... ");
  Serial.println(ssid);
  WiFi.begin(ssid, password); // Enviar los parámetros del ssid y password para su conexión
  WiFi.config(staticIP, gateway, subnet); // Enviar los parámetros de staticIP, gateway y subnet para su configuración

  while (WiFi.status() != WL_CONNECTED) { // Definir un tiempo de espera para la conexión
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi Conectado.... ");
  server.begin(); // Inicializamos el servidor si la conexión es exitosa
  Serial.println("Servidor inicializado");

  Serial.print("Use esta URL para la conexión: ");
  Serial.print("http://");
  Serial.print(WiFi.localIP()); // Imprimir la IP asignada
  Serial.println("/");
}
```

Figura 30 Esta figura muestra el resultado y configuración del módulo de comunicación ESP8266.

Elaborado por: Los Autores

Posteriormente, para el desarrollo de la interfaz Web se usará la herramienta de Visual Studio Code, dicha herramienta se caracteriza por ser amigable y por contar con soporte nativo para lenguajes de diseño Web, como resultado se podrá trabajar de una manera fluida y eficiente. Además, se podrá reducir el consumo de recursos en el ordenador, característica principal de esta herramienta. De este modo se obtendrá la interfaz desarrollada la cual se detalla a continuación.

6.4 Interfaz Desarrollada

La interfaz del simulador cuenta con: una barra de menú, botones controladores, salidas de tipo texto, contenedores, botones de selección para sonidos y un reproductor de voces. La interfaz desarrollada se puede observar en la Figura 31.

Interfaz Web del simulador

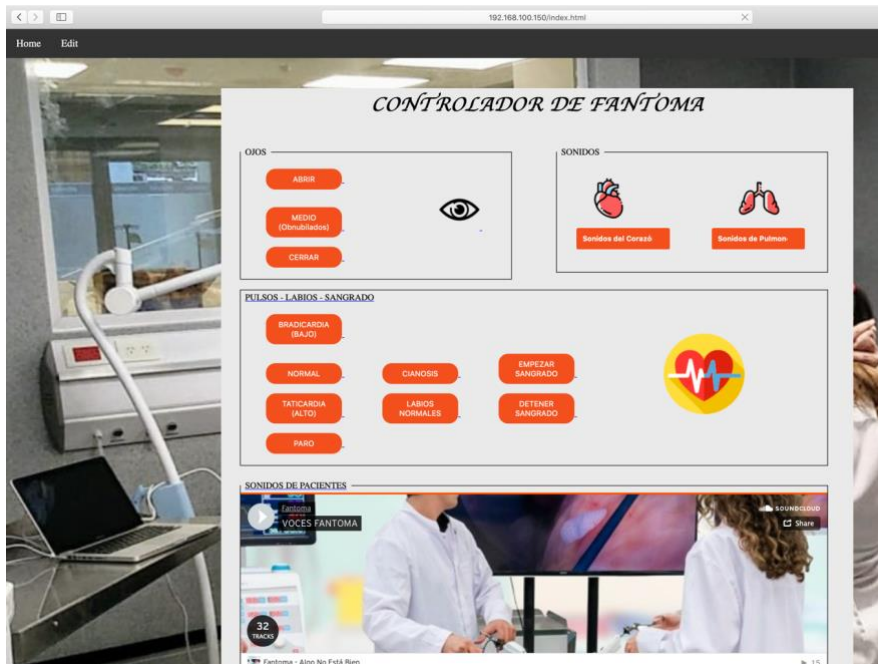


Figura 31 Esta figura muestra la interfaz Web donde el usuario podrá controlar al simulador.
Elaborado por: Los Autores

Todos los elementos anteriormente mencionados se especificarán de mejor manera en los siguientes apartados.

6.4.1 Herramientas del controlador.

La página Web cuenta con una barra de menú que permite al usuario descargarse las credenciales de acceso a SoundCloud donde podrá gestionar los diferentes sonidos que la paciente pueda tener, tal como se muestra en la Figura 32.

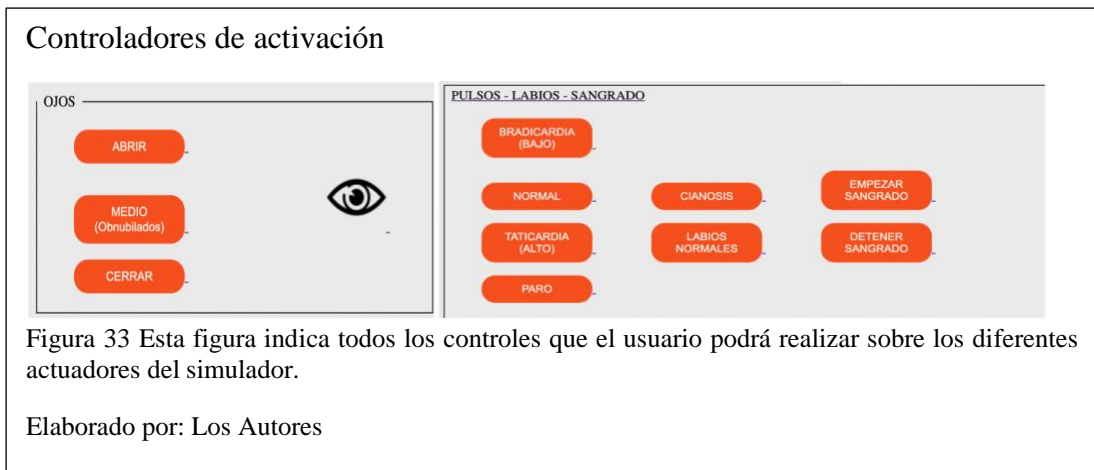
Barra de herramientas



Figura 32 Esta figura muestra las herramientas que podrá usar el usuario para gestionar los sonidos de la paciente.
Elaborado por: Los Autores

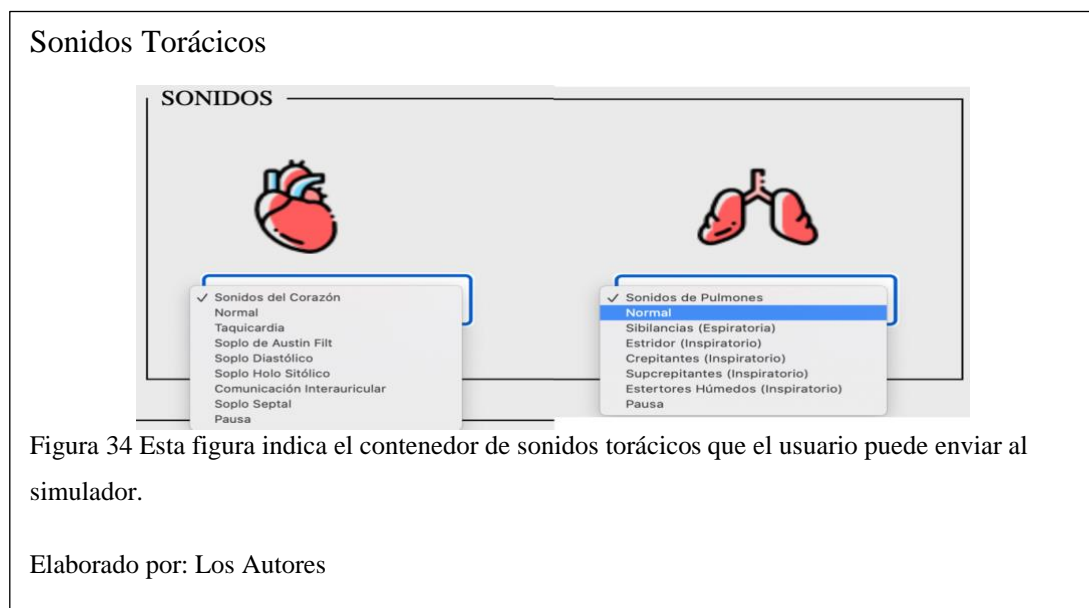
6.4.2 Contenedores de control para activación del simulador

Como muestra la Figura 33, la página cuenta con 2 contenedores de control que permiten el envío de parámetros al simulador, cada uno de ellos hace referencia a los diferentes actuadores tales como: ojos, labios, pulsos cardíacos y bomba de sangre.



6.4.3 Contenedor de sonidos torácicos

El usuario tendrá la posibilidad de seleccionar las diferentes complicaciones que el simulador reproducirá tanto del corazón como de los pulmones, a través de un menú desplegable tal como se puede observar en la Figura 34.



6.4.4 Contenedor de sonidos de la paciente

Tal como se aprecia en la Figura 35 todos los sonidos que tendrá la paciente deberán ser cargados previamente en la plataforma SoundCloud, esto lo realizará la persona encargada de la clínica o el docente a cargo.

Sonidos de la Paciente



The image shows a SoundCloud interface for a playlist titled "SONIDOS DE PACIENTES". The main visual is a video player with a waveform overlay, showing a play button and a "Share" button. Below the player is a list of tracks:

- Fantoma - Algo No Esta Bien (15)
- Fantoma - Buenos Días Dr. Mi Nombre Es Kathy Rojas (9)
- Fantoma - Desde Esta Mañana (4)
- Fantoma - Doctor Por Favor Ayudeme (4)
- Fantoma - Dr. Mi Familia Está Afuera Debe Estar Muy Preocupada Comuniquete Como Me Encuentro (4)
- Fantoma - Es Muy Fuerte

Cookie policy

Figura 35 Esta figura indica el contenedor de sonidos que la paciente podrá emitir dependiendo de la complicación médica.

Elaborado por: Los Autores

CAPÍTULO VII

RESULTADOS

En el presente capítulo se evidencia los resultados realizados a través de pruebas de aceptación correspondientes a cada una de las iteraciones e historias presentadas anteriormente, en este punto fue fundamental contar con la colaboración del ingeniero a cargo de la clínica de simulación quien, validó el correcto funcionamiento del simulador, a continuación, se presenta evidencia de dichas pruebas.

7.1 Pruebas de Aceptación

Con el objetivo de avalar el correcto funcionamiento del sistema se realizan todas las pruebas necesarias referentes a cada una de las iteraciones detalladas anteriormente en el capítulo 5, esto permitió validar correctamente todos los requerimientos solicitados por el usuario, los mismos que se detallan a continuación y que a su vez serán verificados por el ingeniero a cargo de la clínica de simulación.

7.1.1 Pruebas de la primera iteración

Tal como se muestra en la Tabla 47, se cuenta con dos historias de usuario que se asignaron a la primera iteración, a las cuales se les asignó un código el mismo que permite identificar la prueba a realizar.

Tabla 47 Caso de pruebas primera iteración

Código de Prueba	Código de Iteración	Nombre de Prueba
CP1	PI1	Acceso al Sistema
CP2	PI2	Gestión de credenciales multimedia

Nota. Esta tabla muestra la asignación de códigos de prueba a cada una de las historias de usuario en la primera iteración.

Elaborado por: Los Autores

7.1.1.1 Descripción de pruebas

La Tabla 48 hace referencia al caso de prueba correspondiente al acceso del sistema, donde se detalla el proceso de ejecución, pasos de ejecución y los resultados esperados. Este proceso es comprobado por el ingeniero a cargo de la clínica de simulación tal como se observa en la Figura 36.

Tabla 48 Caso de prueba acceso al sistema

CASO DE PRUEBA	
Código: CP1	N° de Historia: PI1
Historia de Usuario: Acceso al Sistema	
Condiciones de Ejecución: Cada docente debe conocer la IP del simulador y verificar que la computadora cuente con acceso a la red de la clínica de simulación.	
Pasos de Ejecución: <ul style="list-style-type: none">• Abrir el navegador de preferencia.• Ingresar en la parte superior la IP del simulador	
Resultado Esperado: Acceso al controlador del Sistema	
Evaluación de la prueba: La prueba terminó exitosamente	

Nota. Esta tabla muestra las pruebas realizadas referentes al acceso del sistema realizadas por el ingeniero a cargo de la clínica de simulación.

Elaborado por: Los Autores

Caso de prueba acceso al sistema

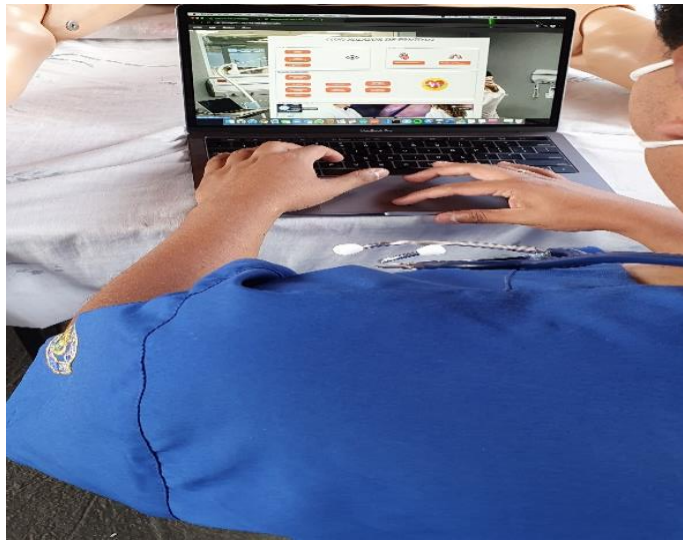


Figura 36 Esta figura muestra la correcta validación del caso de prueba acceso al sistema por parte del ingeniero a cargo de la clínica de simulación.

Elaborado por: Los Autores

La Tabla 49 hace referencia al caso de prueba correspondiente a la gestión de credenciales multimedia, donde se detalla el proceso de ejecución, pasos de ejecución y los resultados esperados.

Tabla 49 Caso de prueba gestión de credenciales multimedia

CASO DE PRUEBA	
Código: CP2	N° de Historia: PI2
Historia de Usuario: Gestión de credenciales multimedia	
Condiciones de Ejecución: Antes de subir o eliminar algún sonido es necesario que el docente cuente con las credenciales correspondientes del SoundCloud.	
Pasos de Ejecución:	
<ul style="list-style-type: none">• Dar clic en el botón Edit ubicado en la barra de menú.• Dar clic a la opción Credenciales donde permitirá descargarse el archivo con el usuario y contraseña de SoundCloud	

- Una vez que se cuente con las credenciales es necesario dar clic en la opción de **Cargar Sonidos**, el cual re direcciona a la página de SoundCloud.
- En la página principal de SoundCloud, dar clic en **Iniciar Sesión** e ingresar el usuario y contraseña.

Resultado Esperado: Acceso a la gestión de los sonidos de la paciente

Evaluación de la prueba: La prueba termino exitosamente.

Nota. Esta tabla muestra las pruebas realizadas referentes a la gestión de credenciales multimedia realizadas por el ingeniero a cargo de la clínica de simulación.

Elaborado por: Los Autores

7.1.2 Pruebas de la segunda iteración

Tal como se muestra en la Tabla 50, se cuenta con dos historias de usuario que se asignaron a la segunda iteración, a las cuales se les asigno un código el mismo que permite identificar la prueba a realizar.

Tabla 50 Caso de pruebas segunda iteración

Código de Prueba	Código de Iteración	Nombre de Prueba
CP3	SI1	Gestión de movimiento de ojos
CP4	SI2	Gestión labios

Nota. Esta tabla muestra la asignación de códigos de prueba a cada una de las historias de usuario en la segunda iteración.

Elaborado por: Los Autores

7.1.2.1 Descripción de pruebas

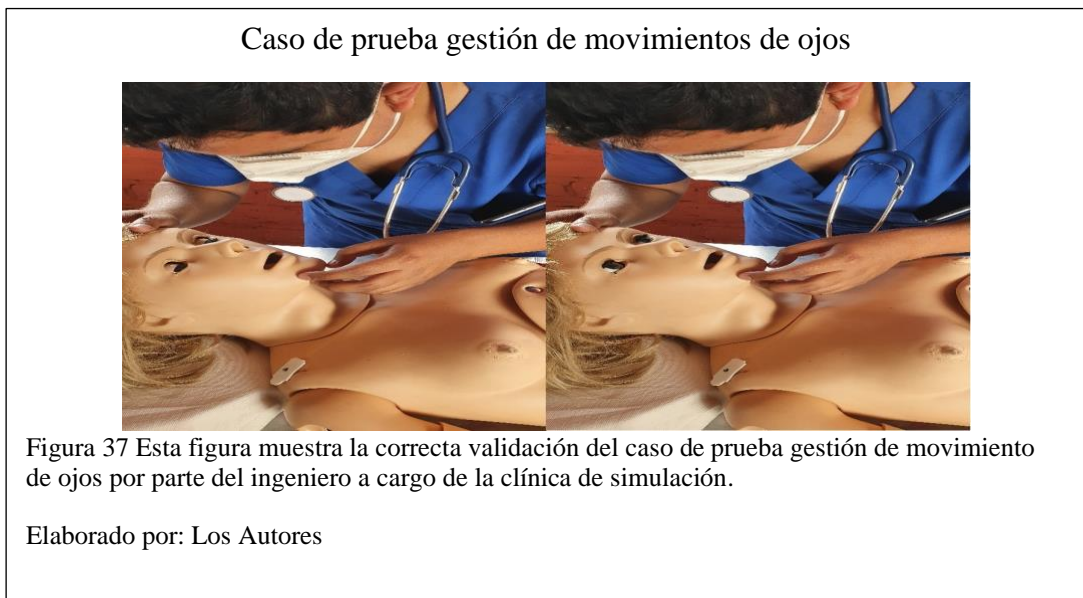
La Tabla 51 hace referencia al caso de prueba correspondiente a la gestión de movimiento de ojos, donde se detalla el proceso de ejecución, pasos de ejecución y los resultados esperados. Este proceso es comprobado por el ingeniero a cargo de la clínica de simulación tal como se observa en la Figura 37.

Tabla 51 Caso de prueba gestión de movimiento de ojos

CASO DE PRUEBA	
Código: CP3	N° de Historia: SI1
Historia de Usuario: Gestión de movimiento de ojos	
Condiciones de Ejecución: Antes de enviar cualquier movimiento es necesario refrescar la página para constatar que la conexión no se haya perdido.	
Pasos de Ejecución:	
<ul style="list-style-type: none"> • Dirigirse a la sección de ojos en la pantalla principal • Dar clic a cualquiera de los 2 movimientos • Revisar el cambio seleccionado en el simulador 	
Resultado Esperado: Cambio visualizados en el simulador	
Evaluación de la prueba: La prueba termino con éxito	

Nota. Esta tabla muestra las pruebas realizadas referentes a la gestión de movimiento de ojos realizadas por el ingeniero a cargo de la clínica de simulación.

Elaborado por: Los Autores



La Tabla 52 hace referencia al caso de prueba correspondiente a la gestión de labios, donde se detalla el proceso de ejecución, pasos de ejecución y los resultados

esperados. Este proceso es comprobado por el ingeniero a cargo de la clínica de simulación tal como se observa en la Figura 38.

Tabla 52 Caso de prueba gestión de labios

CASO DE PRUEBA	
Código: CP4	N° de Historia: SI2
Historia de Usuario: Gestión de labios	
Condiciones de Ejecución: Antes de enviar cualquier movimiento es necesario refrescar la página para constatar que la conexión no se haya perdido	
<ul style="list-style-type: none"> • Dirigirse a la sección de pulsos cardíacos en la pantalla principal • Dar clic a cualquiera de los 2 estados de labios sea cianosis o normales • Revisar el cambio seleccionado en el simulador 	
Resultado Esperado: Cambio visualizado en el simulador	
Evaluación de la prueba: La prueba termino exitosamente.	

Nota. Esta tabla muestra las pruebas realizadas referentes a la gestión de labios realizadas por el ingeniero a cargo de la clínica de simulación.

Elaborado por: Los Autores

Gestión de movimiento de labios




Figura 38 Esta figura muestra la correcta validación del caso de prueba gestión de movimiento de labios por parte del ingeniero a cargo de la clínica de simulación.

Elaborado por: Los Autores

7.1.3 Pruebas de la tercera iteración

Tal como se muestra en la Tabla 53, se cuenta con dos historias de usuario que se asignaron a la tercera iteración, a las cuales se les asignó un código el mismo que permite identificar la prueba a realizar.

Tabla 53 Caso de pruebas tercera iteración

Código de Prueba	Código de Iteración	Nombre de Prueba
CP5	TI1	Gestión de pulsos cardíacos
CP6	TI2	Gestión de sonidos torácicos
CP7	TI3	Gestión de bomba de sangre

Nota. Esta tabla muestra la asignación de códigos de prueba a cada una de las historias de usuario en la tercera iteración.

Elaborado por: Los Autores

7.1.3.1 Descripción de pruebas

La Tabla 54 hace referencia al caso de prueba correspondiente a la gestión de pulsos cardíacos, donde se detalla el proceso de ejecución, pasos de ejecución y los resultados esperados. Este proceso es comprobado por el ingeniero a cargo de la clínica de simulación tal como se observa en la Figura 39.

Tabla 54 Caso de prueba gestión de pulsos cardíacos

CASO DE PRUEBA	
Código: CP5	N° de Historia: TI1
Historia de Usuario: Gestión de pulsos cardíacos	
Condiciones de Ejecución: Antes de enviar cualquier movimiento es necesario refrescar la página para constatar que la conexión no se haya perdido, de igual manera es necesario contar con el estetoscopio.	
Pasos de Ejecución: <ul style="list-style-type: none">• Dirigirse a la sección de pulsos cardíacos en la pantalla principal	

- Dar clic a cualquiera de los 4 pulsos.
- Escuchar el cambio seleccionado en el simulador

Resultado Esperado: Cambio reflejado en el simulador

Evaluación de la prueba: La prueba termino con éxito

Nota. Esta tabla muestra las pruebas realizadas referentes a la gestión de movimiento de pulsos cardíacos realizadas por el ingeniero a cargo de la clínica de simulación

Elaborado por: Los Autores

Caso de prueba pulsos cardíacos



Figura 39 Esta figura muestra la correcta validación del caso de prueba gestión de pulsos cardíacos por parte del ingeniero a cargo de la clínica de simulación.

Elaborado por: Los Autores

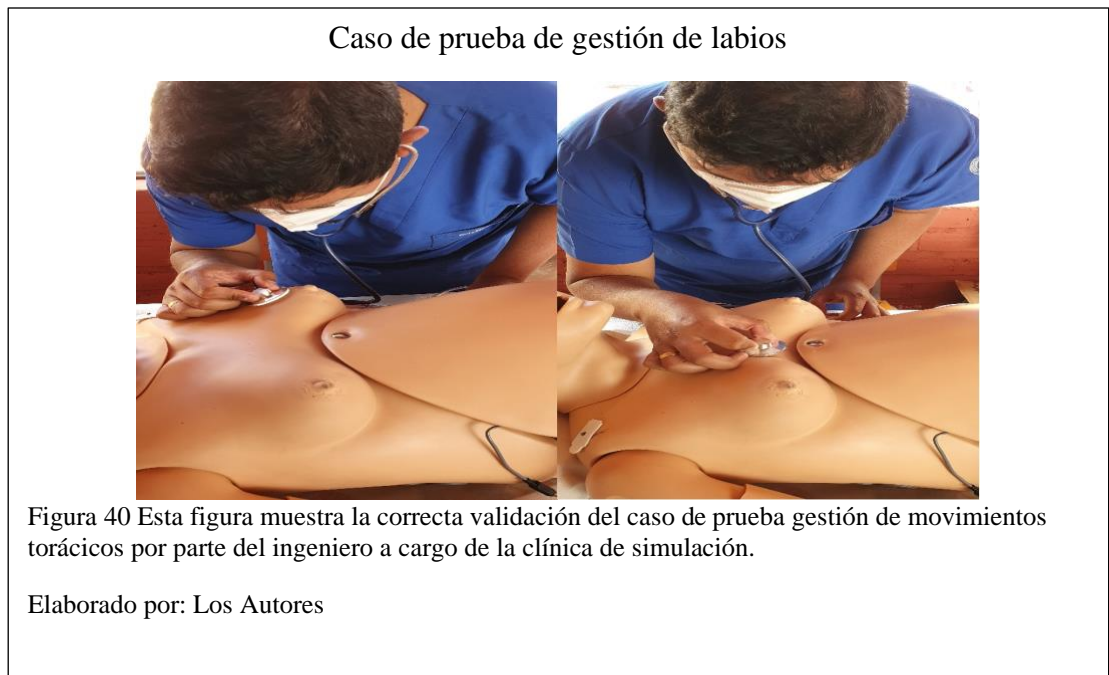
La Tabla 55 hace referencia al caso de prueba correspondiente a la gestión de sonidos torácicos, donde se detalla el proceso de ejecución, pasos de ejecución y los resultados esperados. Este proceso es comprobado por el ingeniero a cargo de la clínica de simulación tal como se observa en la Figura 40.

Tabla 55 Caso de prueba gestión de movimientos torácicos

CASO DE PRUEBA	
Código: CP6	N° de Historia: TI2
Historia de Usuario: Gestión de sonidos torácicos	
Condiciones de Ejecución: Antes de enviar cualquier movimiento es necesario refrescar la página para constatar que la conexión no se haya perdido, de igual manera es necesario contar con el estetoscopio.	
<ul style="list-style-type: none"> • Dirigirse a la sección de sonidos en la pantalla principal • Dar clic a cualquiera de los 2 sonidos sea corazón o pulmones • Seleccionar cualquier síntoma de la lista de sonidos • Escuchar el cambio seleccionado en el simulador 	
Resultado Esperado: Cambio reflejado en el simulador	
Evaluación de la prueba: La prueba termino exitosamente.	

Nota. Esta tabla muestra las pruebas realizadas referentes a la gestión de labios realizada por el ingeniero a cargo de la clínica de simulación.

Elaborado por: Los Autores



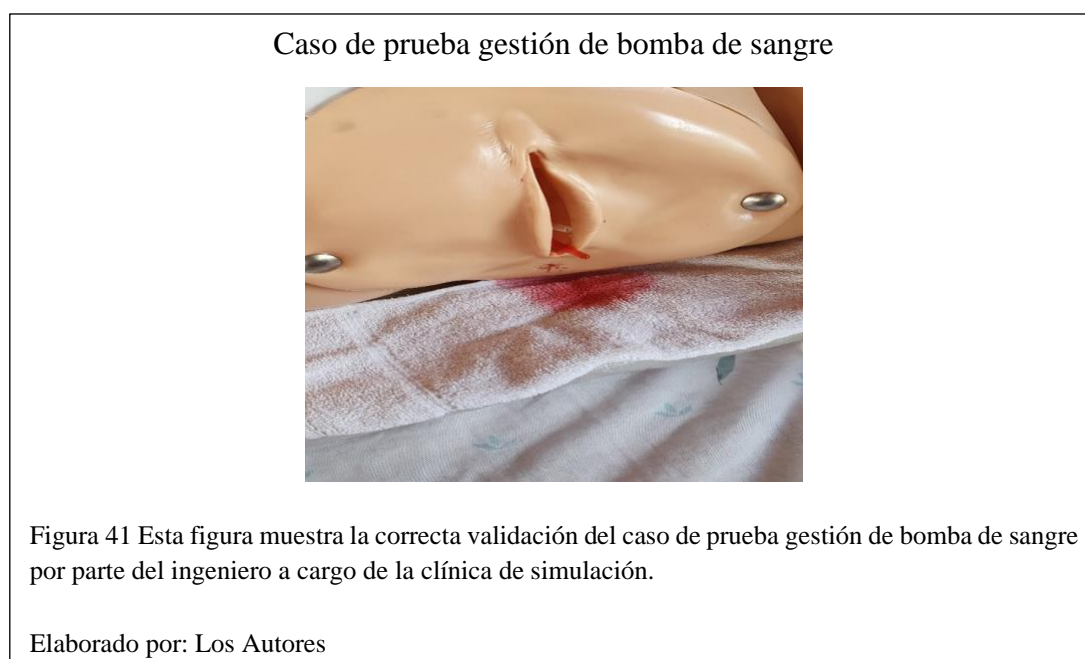
La Tabla 56 hace referencia al caso de prueba correspondiente a la gestión de la bomba de sangre, donde se detalla el proceso de ejecución, pasos de ejecución y los resultados esperados. Este proceso es comprobado por el ingeniero a cargo de la clínica de simulación tal como se observa en la Figura 41.

Tabla 56 Caso de prueba gestión de bomba de sangre

CASO DE PRUEBA	
Código: CP7	N° de Historia: TI3
Historia de Usuario: Gestión de bomba de sangre	
Condiciones de Ejecución: Antes de enviar cualquier movimiento es necesario refrescar la página para constatar que la conexión no se haya perdido, de igual forma es necesario revisar que la manguera de la bomba esté conectada a la funda de sangre principal.	
<ul style="list-style-type: none"> • Dirigirse a la sección de pulsos cardíacos en la pantalla principal • Dar clic a cualquiera de los 2 estados de sangrado • Revisar el cambio seleccionado en el simulador 	
Resultado Esperado: Cambio visualizado en el simulador	
Evaluación de la prueba: La prueba termino exitosamente.	

Nota. Esta tabla muestra las pruebas realizadas referentes a la gestión de bomba de sangre realizadas por el ingeniero a cargo de la clínica de simulación.

Elaborado por: Los Autores



7.2 Validación del Proyecto.

Indicador	Parámetro	Ing. David Erazo
Ojos	Abiertos	✓
	Obnubilados	✓
	Cerrados	✓
Labios	Cianosis	✓
Corazón	Taquicardia	✓
	Soplo Diastólico	✓
	Soplo de Austin Flint	✓
	Soplo Holo sistólico	✓
	Soplo Diastólico	✓
	Soplo Septal	✓
Pulmones	Sibilancias (Espiratoria)	✓
	Estridor (Inspiratorio)	✓
	Crepitantes (Inspiratorio)	✓
	Subcrepitantes (Inspiratorio)	✓
	Estertores Húmedos	✓
Pulsos	Bradycardia	✓
	Taquicardia	✓
	Sin Pulso	✓
Bomba de Sangre		✓
Sonidos de la Paciente		✓

Nota. Esta tabla muestra todas las validaciones individuales del simulador realizada por el Ingeniero David Erazo encargado de la clínica de simulación.

Elaborado por: Los Autores

CONCLUSIONES

Una vez realizado el actual trabajo, se puede confirmar que se han alcanzado los objetivos marcados.

Con la reingeniería de la aplicación se logró mejorar la estabilidad del sistema, además de la velocidad de la conexión, un punto a favor que ayudara en las diversas prácticas que los estudiantes desarrollaran.

De igual manera, mediante la investigación realizada para comparar el módulo bluetooth que se encontraba instalado en el simulador con el modulo actualmente integrado se verificó que el tiempo de respuesta al ejecutar una acción mejora considerablemente, esto se debe a que la potencia de transmisión es mayor permitiendo que la transferencia de paquetes sea menor a los 2 ms.

La creación del sistema informático permitió controlar completamente todas las funciones del simulador; por lo que, no necesita ningún tipo de instalación o configuración de puertos, ya que la misma se encuentra integrada en el módulo Wi-Fi y funciona únicamente al digitar la dirección IP del simulador, obteniendo de esta manera un software confiable e intuitivo al momento de usarlo.

RECOMENDACIONES

En este trabajo se utilizó un módulo Arduino, sin embargo, este dispositivo es limitado por lo tanto para un mejor desempeño o ampliación del sistema se recomienda utilizar un dispositivo más robusto como por ejemplo Raspberry Pi.

Una vez usada la bomba de sangre en la fantoma se recomienda quitar todo el líquido dentro de la misma, esto evitara que la bomba se quemara y pueda ocasionar futuros problemas.

REFERENCIAS

Artículos Académicos

- Carriel, J., & Zambrano, L. (2014). Centro de simulación médica de la Universidad Católica de Santiago de Guayaquil: Una propuesta de planificación estratégica. *Guayaquil: Universidad Católica de Santiago de Guayaquil.*
- Aroca, R. V, Caurin, G., & Carlos-SP-Brasil, S. (2009). A real time operating systems (RTOS) comparison. *WSO-Workshop de Sistemas Operacionais, 12.*
- Pérez, H. P. (2003). Sistemas Operativos de tiempo real (RTOS). *Revista Española de Electrónica, 580*, 42–43.
- Ramonedá, V. C., & Mussons, F. B. (2008). Preeclampsia. Eclampsia y síndrome HELLP. *Protocolos Diagnóstico Terapéuticos de La AEP. Neonatología.*
- Russell, D. J. (2010). Introduction to embedded systems: using ANSI C and the arduino development environment. *Synthesis Lectures on Digital Circuits and Systems, 5*(1), 1–275.
- Solari, A. A., Solari, G. C., Wash, F. A., Guerrero, G. M., & Enríquez, G. O. (2014). Hemorragia del postparto. Principales etiologías, su prevención, diagnóstico y tratamiento. *Revista Médica Clínica Las Condes, 25*(6), 993–1003.
- Vignaga, A., & Perovich, D. (2008). Arquitecturas y tecnologías para el desarrollo de aplicaciones Web. *Montevideo, Uruguay.*
- Vitores, R. S. (2004). Aplicaciones de los sistemas embebidos. *Técnica Industrial, 1*, 24–27.
- Ye, L. (2020). Study on embedded system in monitoring of intelligent city pipeline network. *Computer Communications, 153*, 451–458.

Bibliografía

- Arriarán, S. S. (2017). *Todo sobre sistemas embebidos: Arquitectura, programación y diseño de aplicaciones prácticas con el PIC18F*. Universidad Peruana de Ciencias Aplicadas.
- Artero, Ó. T. (2013). *ARDUINO. Curso práctico de formación*. RC Libros.
- García, J. J., Fernández, J. L., & Montoya, F. J. (2005). Introducción a la Programación. *Un Enfoque Algorítmico*. Thomson, 8.
- Heath, S. (2002). *Embedded systems design*. Elsevier.
- Herrador, S. G. (2010). *HTML & CSS Fácil y sencillo*.
- Mohedano, J., Saiz, J. M., & Román, P. S. (2012). *Iniciación a javascript*. Ministerio de Educación.
- Vértice, E. (2009). *Diseño básico de páginas Web en HTML*. Editorial Vértice.
- Hernández Camacho, H. (2016). *EJE 3: Electrónica y Control*. Guadalajara: Universidad de Guadalajara.

Conferencias

- Badamasi, Y. A. (2014). The working principle of an Arduino. *2014 11th International Conference on Electronics, Computer and Computation (ICECCO)*, 1–4.
- Camargo, C., Cortés, J., & Jiménez, A. (2012). Implementación de sistemas digitales complejos utilizando sistemas embebidos. *Ingenium Revista de La Facultad de Ingeniería*, 13(25), 5–15.
- Carriel, J., & Zambrano, L. (2014). Centro de simulación médica de la Universidad Católica de Santiago de Guayaquil: Una propuesta de planificación estratégica. *Guayaquil: Universidad Católica de Santiago de Guayaquil*.
- Carty, D. M., Delles, C., & Dominiczak, A. F. (2010). Preeclampsia and future

maternal health. *Journal of Hypertension*, 28(7), 1349–1355.

Carvajal, J., & Ralph, C. (2017). Manual de Obstetricia y Ginecología. *Pontificia Católica de Chile*, 7.

Chyliński, M., & Szmajda, M. (2019). Design and Implementation of an Embedded System for Respiratory Rate Examinations. *IFAC-PapersOnLine*, 52(27), 341–346.

del Rey, J., de Aldana, E., & others. (2014). *La localización de Webs dinámicas: objetos, métodos, presente y futuro*.

Gardner, R., & Raemer, D. B. (2008). Simulation in obstetrics and gynecology. *Obstetrics and Gynecology Clinics of North America*, 35(1), 97–127.

Greif, D., Bottaro, S., Gómez, F., Grenno, A., Nozar, F., Fiol, V., & Briozzo, L. (2015). Capacitación de residentes de ginecología en urgencias obstétricas mediante simulación clínica. *Revista Médica Del Uruguay*, 31(1), 46–52.

Hongxing, W., & Tianmiao, W. (2006). Curriculum of Embedded System for Software Colleges. *2006 2nd IEEE/ASME International Conference on Mechatronics and Embedded Systems and Applications*, 1–5.

Ruiz-Berdún, D., & Blanco, A. G. (2014). Modelos Para la Enseñanza de la Obstetricia a lo Largo de la Historia. *Enseñanza e Historia de Las Ciencias y de Las Técnicas: Orientación, Metodología y Perspectivas: Actas Del VII Simposio de Enseñanza e Historia de Las Ciencias y de Las Técnicas de La SEHCYT, (Barcelona, Institut d'Estudis Catalans, 15-16 Marzo 2013)*, 121–125.

Imágenes

Medical Expo. (12 de 02 de 2020). *Simulador de palpación de embarazo*. Obtenido de <https://www.medicaexpo.es/prod/health-edco/product-119758-834813.html>

Sitios Web

Ferrer, J., García, V., & García, R. (2013). Curso completo de HTML. *Recuperado de: <Http://Es.Tldp.Org/Manuales-LuCAS/Doc-Curso-Html/Doc-Curso-Html.Pdf>*.

Marini, E. (2012). El modelo cliente/servidor. *Recuperado de:*

<https://www.linuxito.com/docs/el-modelo-cliente-servidor.pdf>

Murcia Esquivá, J. L. (2018). *Sistema watchdog digital configurable.*

paraarduino. (2020). *Modelos de placas Arduino en 2020. Obtenido de*

<https://paraarduino.net/modelos-de-placas-arduino>

Rodríguez-Palchevich, D.-R. (2008). *Nuevas tecnologías Web 2.0: Hacia una real democratización de la información y el conocimiento. Recuperado de:*

http://biblioteca.udgvirtual.udg.mx/jspui/bitstream/123456789/3564/1/Nuevas_tecnolog%c3%adas_Web_2.0.pdf

Velasco, N. (2005). Sistema embebido para la conexión de un PLC Siemens S7-200 a la red GSM. *Obtenido de E-REDING. Biblioteca de La Escuela Superior de Ingenieros de Sevilla: Disponible En Web*

<Http://Bibing.Us.Es/Proyectos/Abreproy/11141/Fichero/PFC%252F0+portada+e+%C3%ADndices.Pdf+>.

Tesis

Almaraz Hernández, J. M., Campos Cantero, P., & Castelo Delgado, T. (2011). *Desarrollo de una aplicación Web para la gestión de Entornos Virtuales. Trabajo de Titulación previo a la obtención del Título de Ingenieros en Sistemas Informáticos.*

Bonillo, A. L. (2004). *Comunicación Punto a Punto vía modem GSM. Tesis de*

pregrado, Universidad Rovira i Virgili.

Camargo Bareño, C. I. (2011). *Transferencia tecnológica y de conocimientos en el diseño de sistemas embebidos*. Universidad Nacional de Colombia.

Díaz Mulas, B. (2015). *UART: Universal Asynchronous Receiver-Transmitter*.

Falcón Ayala, J. F. (2016). *Funcionamiento mantenimiento y reposición del equipamiento médico fantomas de la Clínica de Simulación Médica y Robótica de la Facultad de Ciencias Médicas de la Universidad Central del Ecuador. Trabajo de Titulación previo a la obtención del Título de Licenciado en Electromedicina. Carrera de Electromedicina*. Quito: UCE.

Heredia Iza, P. F., & Montaluisa Montaluisa, A. D. (2020). *Prototipo electrónico para controlar y programar escenarios clínicos en un simulador gineco-obstétrico de baja fidelidad. Trabajo de Titulación previo a la obtención del Título de Ingenieros Electrónicos*. Universidad Politécnica Salesiana.

Márquez Avendaño, B., & Zulaica Rugarcía, J. (2004). Implementación de un reconocedor de voz gratuito a el sistema de ayuda a invidentes Dos-Vox en español. *Trabajo Fin de Carrera, Universidad de Las Américas Puebla*.

ANEXOS

Anexo 1

Código de configuración del módulo ESP8266

```
#include <ESP8266WiFi.h>
#include <SoftwareSerial.h>

const char* ssid = "Sergio_2.4G";
const char* password = "1721326559.SERGIO.1994";
SoftwareSerial s(3,1);

IPAddress staticIP(192,168,100,150); // Declarar una IP estática dentro de la red
IPAddress gateway(192,168,100,1); // Declarar la puerta de enlace
IPAddress subnet(255,255,255,0); // Declarar la máscara de red
WiFiServer server(80);
void setup() {
  s.begin(9600);
  //Serial.begin(115200);

  // Connect to WiFi network
  Serial.print("Connecting to..... ");
  Serial.println(ssid);

  WiFi.begin(ssid, password); // Enviar los parámetros del ssid y password para su conexión
  WiFi.config(staticIP, gateway, subnet); // Enviar los parámetros de staticIP, gateway y subnet para su configuración

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");

  // Start the server
  server.begin();

  // Print the IP address

}

void loop() {
  // Check if a client has connected
  WiFiClient client = server.available();
  if (!client) {
    return;
  }
}
```

```

// Wait until the client sends some data
Serial.println("new client");
while(!client.available()){
  delay(1);
}

// Read the first line of the request
String request = client.readStringUntil('\r');
Serial.println(request);
client.flush();

// Match the request

if (request.indexOf("/abiertos") != -1) {
  s.write('b');
}

if (request.indexOf("/cerrados") != -1) {
  s.write('a');
}

if (request.indexOf("/medios") != -1) {
  s.write('c');
}

if (request.indexOf("/empezar_sangrado") != -1) {
  s.write('x');
}

if (request.indexOf("/detener_sangrado") != -1) {
  s.write('z');
}

if (request.indexOf("/cianocis") != -1) {
  s.write('w');
}

if (request.indexOf("/labios_normales") != -1) {
  s.write('y');
}

```

```

if (request.indexOf("/bradicardia") != -1) {
    s.write('d');
}

if (request.indexOf("/normal") != -1) {
    s.write('e');
}

if (request.indexOf("/taticardia") != -1) {
    s.write('f');
}

if (request.indexOf("/paro") != -1) {
    s.write('g');
}

if (request.indexOf("/Normal") != -1) {
    s.write('p');
}
if (request.indexOf("/Sibilanciasia") != -1) {
    s.write('q');
}
if (request.indexOf("/Estridor") != -1) {
    s.write('r');
}
if (request.indexOf("/Crepitantes") != -1) {
    s.write('s');
}
if (request.indexOf("/Supcrepitantes") != -1) {
    s.write('t');
}
if (request.indexOf("/Estertores") != -1) {
    s.write('u');
}
if (request.indexOf("/Pausa_Pulmon") != -1) {
    s.write('v');
}

if (request.indexOf("/Normal_Corazon") != -1) {
    s.write('h');
}
if (request.indexOf("/Taquicardia") != -1) {
    s.write('i');
}

if (request.indexOf("/Soplo_de_Austin") != -1) {
    s.write('j');
}
if (request.indexOf("/Soplo_Diastólico") != -1) {
    s.write('k');
}
if (request.indexOf("/Soplo_Holo_Sitólico") != -1) {
    s.write('l');
}
if (request.indexOf("/Comunicación_Interauricular") != -1) {
    s.write('m');
}
if (request.indexOf("/Soplo_Septal") != -1) {
    s.write('n');
}
if (request.indexOf("/Pausa_Corazon") != -1) {
    s.write('o');
}
}

```

Anexo 2

Código de la página Web para el control del simulador.

```
// Return the response
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println(""); // do not forget this one
client.println("<!DOCTYPE HTML>");
client.println("<html>");
client.println("<head>");
client.println("<title>SIMULACION | FANTOMA</title>");
client.println("<meta charset='UTF-8'>");
client.println("<meta name='viewport' content='width=device-width, initial-scale=1.0'>");
//INICIO SCRIPTS

//fin script
client.println("</head>");
//ESTILOS
client.println("<style type='text/css'>");
client.println("body {");
client.println("background-image: url('https://i.ibb.co/h9kTBH1/fondo3.png');");
client.println("}");

client.println("*{");
client.println("margin: 0px;");
client.println("padding: 0px;");
client.println("background-size: cover;");
client.println("}");

client.println(".contenedor input[type='text']{");
client.println("border: none;");
client.println("border-bottom: 2px solid black; ");
client.println("background: transparent;");
client.println("outline: none;");
client.println("width: 230px;");
client.println("height: 30px;");
client.println("color: black;");
client.println("font-size: 14px;");
client.println("}");

client.println("div#contenedor{");
client.println("margin: auto;");
client.println("margin-top: 30px;");
client.println("width: 1000px;");
client.println("height: 1000px;");
client.println("background-color: #eaeaea;");
client.println("}");
```

```
client.println("div#cabezera{");
client.println("width: 1000px;");
client.println("height: 60px;");
client.println("float: top;");
client.println("background-color: #eaeaea;");
client.println("}");
```

```
client.println("div#contenedor_sonidos{");
client.println("float: right;");
client.println("width: 430px;");
client.println("height: 200px;");
client.println("margin-top: 30px;");
client.println("margin-right: 40px;");
client.println("}");
```

```
client.println("div#contenedor_ojos{");
client.println("width: 430px;");
client.println("height: 200px;");
client.println("margin-top: 30px;");
client.println("margin-left: 30px;");
client.println("}");
```

```
client.println("div#contenedor_pulsos{");
client.println("width: 930px;");
client.println("height: 250px;");
client.println("margin-top: 30px;");
client.println("margin-left: 30px;");
client.println("}");
```

```
client.println("div#img_pulsos{");
client.println("width: 200px;");
client.println("height: 150px;");
client.println("ext-align: center;");
client.println("margin-top: -210px;");
client.println("margin-right: 60px;");
client.println("float: right;");
client.println("}");
```

```
client.println("div#contenedor_reproductor{");
client.println("width: 930px;");
client.println("height: 350px;");
client.println("margin-top: 50px;");
client.println("margin-left: 30px;");
client.println("}");
```

```

client.println("label#imagen_co {");
client.println("display: inline-block;");
client.println("padding: 10px;");
client.println("margin-top: 20px;");
client.println("margin-left: 40px;");
client.println("}");
client.println("label#imagen_pu {");
client.println("display: inline-block;");
client.println("padding: 10px;");
client.println("margin-top: 20px;");
client.println("margin-left: 150px;");
client.println("}");
client.println(" .button {");
client.println(" display: inline-block;");
client.println(" border-radius: 4px;");
client.println("background-color: #f4511e;");
client.println("border: none;");
client.println("border-radius: 15px;");
client.println("color: #FFFFFF;");
client.println("text-align: center;");
client.println("font-size: 12px;");
client.println("padding: 9px;");
client.println("width: 120px;");
client.println("transition: all 0.5s;");
client.println("cursor: pointer;");
client.println("margin-top: 15px;");
client.println("margin-left: 40px;");
client.println("}");

client.println(".button span {");
client.println("cursor: pointer;");
client.println("display: inline-block;");
client.println("position: relative;");
client.println("transition: 0.5s;");
client.println("}");

client.println(".button span:after {");
client.println("content: '\00bb';");
client.println("position: absolute;");
client.println("opacity: 0;");
client.println("top: 0;");
client.println("right: -15px;");
client.println(" transition: 0.6s;");
client.println("}");

```

```

client.println(".button:hover span {}");
client.println("padding-right: 15px;");
client.println("}");
client.println(".button:hover span:after {}");
client.println("opacity: 1;");
client.println("right: 0;");
client.println("}");
client.println("label#imagen_ojos {}");
client.println("display: inline-block;");
client.println(" margin-left: 150px;");
client.println("}");
client.println("fieldset {}");
client.println("border:1px solid black; ");
client.println("}");
client.println("legend {}");
client.println("padding: 0.2em 0.5em;");
client.println("color:black;");
client.println("font-size:90%;");
client.println("text-align:left;");
client.println("}");
client.println(".upload-btn-wrapper {}");
client.println("position: relative;");
client.println("overflow: hidden;");
client.println("display: inline-block;");
client.println("}");
client.println(".btn {}");
client.println("border: 2px solid gray;");
client.println("color: gray;");
client.println("background-color: white;");
client.println("padding: 8px 20px;");
client.println("border-radius: 8px;");
client.println("font-size: 20px;");
client.println("font-weight: bold;");
client.println("}");
client.println(".upload-btn-wrapper input[type=file] {}");
client.println("font-size: 100px;");
client.println("position: absolute;");
client.println("left: 0;");
client.println("top: 0;");
client.println(" opacity: 0;");
client.println("}");

```

```

client.println(".upload-btn-wrapper {");
client.println("position: relative;");
client.println("overflow: hidden;");
client.println("display: inline-block;");
client.println("}");

client.println("h1 {");
client.println("font: oblique bold 200% cursive;");
client.println("text-align: center;");
client.println("}");

client.println(".navbar {");
client.println("overflow: hidden;");
client.println("background-color: #333;");
client.println("}");

client.println(".navbar a {");
client.println("float: left;");
client.println("font-size: 16px;");
client.println("color: white;");
client.println("text-align: center;");
client.println("padding: 14px 16px;");
client.println("text-decoration: none;");
client.println("}");

client.println(".dropdown {");
client.println("float: left;");
client.println("overflow: hidden;");
client.println("}");

client.println(".dropdown .dropbtn {");
client.println("font-size: 16px;");
client.println("border: none;");
client.println("outline: none;");
client.println("color: white;");
client.println("padding: 14px 16px;");
client.println("background-color: inherit;");
client.println("font-family: inherit;");
client.println("margin: 0;");
client.println("}");

client.println(".navbar a:hover, .dropdown:hover .dropbtn {");
client.println("background-color: red;");
client.println("}");

```

```

client.println(".dropdown-content {");
client.println("display: none;");
client.println("position: absolute;");
client.println("background-color: #f9f9f9;");
client.println("min-width: 160px;");
client.println("box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);");
client.println("z-index: 1;");
client.println("}");

client.println(".dropdown-content a {");
client.println("float: none;");
client.println("color: black;");
client.println("padding: 12px 16px;");
client.println("text-decoration: none;");
client.println("display: block;");
client.println("text-align: left;");
client.println("}");
client.println(".dropdown-content a:hover {");
client.println("background-color: #ddd;");
client.println("}");
client.println(".dropdown:hover .dropdown-content {");
client.println("display: block;");
client.println("}");

client.println("select {");
client.println("border-radius: 4px;");
client.println("width: 35%;");
client.println("height: 35px;");
client.println("font-size: 70%;");
client.println("font-weight: bold;");
client.println("cursor: pointer;");
client.println("border-radius: 4;");
client.println("background-color: #f4511e;");
client.println("border-bottom: 2px solid #f4511e;");
client.println("color: white;");
client.println("appearance: none;");
client.println("padding: 10px;");
client.println("padding-right: 28px;");
client.println("-webkit-appearance: none;");
client.println("-moz-appearance: none;");
client.println("transition: color 0.3s ease, background-color 0.3s ease, border-bottom-color 0.3s ease;");
client.println("margin-left: 30px;");
client.println("margin-right: 30px;");
client.println("}");

client.println("select:-ms-expand {");
client.println("display: none;");
client.println("}");

client.println("select:hover,");
client.println("select:focus {");
client.println("color: #000;");
client.println("background-color: white;");
client.println("border-bottom-color: #000;");
client.println("}");
client.println("select:hover - .select-icon,");
client.println("select:focus - .select-icon {");
client.println("background-color: white;");
client.println("border-color: #000;");
client.println("}");

client.println("</style>");
//FIN ESTILOS

//PRINCIPAL BODY
client.println("<body>");
client.println("<script src='https://maxcdn.bootstrapcdn.com/bootstrap/3.1.1/js/bootstrap.min.js'></script>");
client.println("<script src='http://cdnjs.cloudflare.com/ajax/libs/socket.io/1.3.6/socket.io.min.js'></script>");
client.println("<script type='text/javascript'>");
client.println("fondo = new Array(2);");
client.println("fondo[0] = 'https://i.ibb.co/z1nfF6J/fondo2.png'");
client.println("fondo[1] = 'https://i.ibb.co/93jy96J/lago.jpg'");
client.println("fondo[2] = 'https://i.ibb.co/t9k7BTU/fondo3.png'");
client.println("index = Math.floor(Math.random() * fondo.length)");
client.println("document.write('<body BACKGROUND=' + fondo[index] + '>');");

client.println("function validar_abiertas() (var abiertas = document.getElementById('imagen_eyes');abiertas.innerHTML = '<img src='https://i.ibb.co/x3f068/eye.png' alt='Ojos Abiertos'/>';return true;});

client.println("function validar_medios() {");
client.println("var medios = document.getElementById('imagen_eyes');");
client.println("medios.innerHTML = '<img src='https://i.ibb.co/WGfQ54/sleepy-eye.png' alt='Ojos Medios'/>';");
client.println("return true;");
client.println("}");

```



```

client.println("function validar_cerrados() {");
  client.println("var cerrados = document.getElementById('imagen_ojos');");
  client.println("cerrados.innerHTML = '<img src='\"https://i.ibb.co/4ZL25xq/pestona.png\" alt='Ojos Cerrados' />';");
  client.println("return true;");
client.println("}");

client.println("document.getElementById('imagen').onchange = function () {");
  client.println("console.log(this.value);");
  client.println("document.getElementById('fichero').innerHTML = document.getElementById('imagen').files[0].name;");
client.println("}");

client.println("</script>");

client.println("<div class='navbor'><a class='active' href='index.html'>Home</a><div class='dropdown'><button class='dropbtn'>Edit<i class='fa fa-caret-down'></i></button>");
client.println("<a href='\"#\"' id='link'><img src='\"#\"' id='imagen'></a>");
client.println("<div class='contenedor' id='contenedor'>");

client.println("<div id='cabecera'> <h1>CONTROLADOR DE FANTOMA</h1> </div>");

client.println("<div id='contenedor_sonidos'>");
client.println("<fieldset>");
client.println("<legend>SONIDOS</legend>");
client.println("<label id='imagen_co' for='tam' >");
client.println("<img src='\"https://i.ibb.co/VqW6v4/cardiavasculor.png\" alt='CORAZON' />");
client.println("</label>");
client.println("<label id='imagen_pu' for='tam' >");
client.println("<img src='\"https://i.ibb.co/qy61xz8/pulmon.png\" alt='PULMON' />");
client.println("</label>");
client.println("<div>");
//Sonidos del Corazon

client.println("<select onchange='location = this.value;'>");
client.println("<option value='1'>Sonidos del Corazon</option>");
client.println("<a href='\"Normal_Corazon\"'><option value='Normal_Corazon'>Normal</option>");
client.println("<a href='\"Taquicardia\"'><option value='Taquicardia'>Taquicardia</option>");
client.println("<a href='\"Soplo_de_Austin\"'><option value='Soplo_de_Austin'>Soplo de Austin Filt</option>");
client.println("<a href='\"Soplo_Diastolico\"'><option value='Soplo_Diastolico'>Soplo Diastolico</option>");
client.println("<a href='\"Soplo_Halo_Sitolico\"'><option value='Soplo_Halo_Sitolico'>Soplo Halo Sitolico</option>");
client.println("<a href='\"Comunicacion_Interauricular\"'><option value='Comunicacion_Interauricular'>Comunicacion Interauricular</option>");
client.println("<a href='\"Soplo_Septal\"'><option value='Soplo_Septal'>Soplo Septal</option>");
client.println("<a href='\"Pausa_Corazon\"'><option value='Pausa_Corazon'>Pausa</option>");
client.println("</select>");

//Sonidos Pulmonares
client.println("<select onchange='location = this.value;'>");
client.println("<option value='1'>Sonidos de Pulmones</option>");
client.println("<a href='\"Normal\"'><option value='Normal'>Normal</option>");
client.println("<a href='\"Sibilancias\"'><option value='Sibilancias'>Sibilancias (Espiratoria)</option>");
client.println("<a href='\"Estridor\"'><option value='Estridor'>Estridor (Inspiratorio)</option>");
client.println("<a href='\"Crepitantes\"'><option value='Crepitantes'>Crepitantes (Inspiratorio)</option>");
client.println("<a href='\"Subcrepitantes\"'><option value='Subcrepitantes'>Subcrepitantes (Inspiratorio)</option>");
client.println("<a href='\"Estertores\"'><option value='Estertores'>Estertores Húmedos (Inspiratorio)</option>");
client.println("<a href='\"Pausa_Pulmon\"'><option value='Pausa_Pulmon'>Pausa</option>");
client.println("</select>");

client.println("<div>");
client.println("<div>");
client.println("<div>");
client.println("</fieldset>");
client.println("</div>");

client.println("<div id='contenedor_ojos'>");
client.println("<fieldset>");
client.println("<legend>OJOS</legend>");
client.println("<a href='\"abiertos\"'><button class='button' name='favorito' type='button' onclick='return validar_abiertas();'><span>ABRIR</span></button>");
client.println("<div>");
client.println("<a href='\"medios\"'><button class='button' name='favorito' type='button' onclick='return validar_medios();'><span>MEDIO (Obnubilados)</span></button>");
client.println("<label id='imagen_ojos' for='tam' > <img src='\"https://i.ibb.co/x3HTXXB/eye.png\"> </label> ");
client.println("<div>");
client.println("<a href='\"cerrados\"'><button class='button' name='favorito' type='button' onclick='return validar_cerrados();'><span>CERRAR</span></button>");
client.println("<div>");
client.println("<div>");
client.println("<embed/>");
client.println("</fieldset>");
client.println("</div>");

```

Anexo 3

Certificado de validación del funcionamiento del simulador



**Clínica de Simulación
Médica y Robótica**
Universidad Central del Ecuador



Quito Dirección: Iquique y Sodiro s/n Troncal: 2528-690 2528-810 Ext. 160 ó 168 Ecuador

CLINSIM OF N° 175

Quito, 24 de septiembre de 2020

MSc. DAVID ANDRÉS ERAZO CARVAJAL

Coordinador técnico de la Clínica de Simulación Médica y Robótica

Facultad de Ciencias Médicas

Universidad Central del Ecuador

Por medio de la presente hago constar que el proyecto técnico "REDISEÑO Y ELABORACIÓN ELECTRÓNICA PARA CONTROLAR Y PROGRAMAR ESCENARIOS CLÍNICOS EN UN FANTOMA OBSTÉTRICO DE BAJA FIDELIDAD" presentado el día Miércoles 24 de septiembre del 2020 por los estudiantes Angela Dayana Román Cárdenas y Palacios Montenegro Sergio Andrés ha sido validado satisfactoriamente en cada uno de los parámetros de funcionamiento. Por lo tanto se acepta la entrega del prototipo según lo acordado.

Atentamente:

MSc. David Andrés Erazo Carvajal

C.I 1714998489

email: fcm.centro.simulacion@uce.edu.ec

Anexo 4

Tabla de validación de los parámetros del simulador



Tabla de Validación

Fecha: 24 de septiembre del 2020

Validación de complicaciones individuales simuladas en el proyecto técnico "REDISEÑO Y ELABORACIÓN ELECTRÓNICA PARA CONTROLAR Y PROGRAMAR ESCENARIOS CLÍNICOS EN UN FANTOMA OBSTÉTRICO DE BAJA FIDELIDAD" realizado por Sergio Palacios y Angela Román.

Indicador	Parámetro	MSc. David Erazo
Ojos	Abiertos	✓
	Obnubilados	✓
	Cerrados	✓
Labios	Cianosis	✓
Corazón	Taquicardia	✓
	Soplo de Austin Flint	✓
	Soplo Diastólico	✓
	Soplo Holo sistólico	✓
	Comunicación Interauricular	✓
	Soplo septal	✓
Pulmones	Sibilancias (Espiratoria)	✓
	Estridor (Inspiratorio)	✓
	Crepitantes (Inspiratorio)	✓
	Subcrepitantes (Inspiratorio)	✓
	Estertores Húmedos (Inspiratorio)	✓
Pulsos	Bradicardia	✓
	Taquicardia	✓
	Sin Pulso	✓
Bomba de Sangre		✓
Sonidos de la paciente		✓


 MSc. David Erazo