



# ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y  
TELECOMUNICACIONES**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE  
INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES**

**TEMA: DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO PARA  
FISIOTERAPIA CON KINECT**

**AUTOR: FONSECA FACTOS, SONIA ALEJANDRA**

**DIRECTOR: ING. ALULEMA, DARWIN**

**SANGOLQUÍ**

**2016**

*Certificado de tutoría*

**UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE**

**INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES**

**CERTIFICADO**

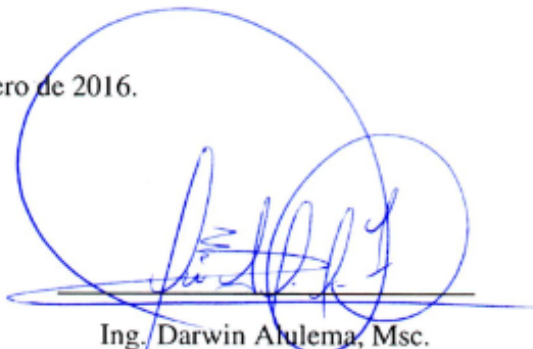
Ing. Darwin Alulema, MSc.

**CERTIFICA**

Que el trabajo titulado “DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO PARA FISIOTERAPIA CON KINECT”, realizado por Sonia Alejandra Fonseca Factos, ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la Universidad de las Fuerzas Armadas - ESPE en su reglamento.

Debido a que se trata de un trabajo de investigación recomiendan su publicación.

Sangolquí, 22 de enero de 2016.



Ing. Darwin Alulema, Msc.  
DIRECTOR

*Declaración de Responsabilidad*

**UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE**

**INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES**

**DECLARACIÓN DE RESPONSABILIDAD**

**SONIA ALEJANDRA FONSECA FACTOS**

**DECLARO QUE:**

El proyecto de grado denominado “DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO PARA FISIOTERAPIA CON KINECT”, ha sido desarrollado en base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme las citas se presentan en las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía.

Consecuentemente este trabajo es de mi autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico del proyecto de grado en mención.

Sangolquí, 22 de enero de 2016.



Sonia Alejandra Fonseca Factos

*Autorización de publicación*

**UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE**

**INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES**

**AUTORIZACIÓN**

**SONIA ALEJANDRA FONSECA FACTOS**

Autorizo a la Universidad de las Fuerzas Armadas - ESPE la publicación, en la biblioteca virtual de la Institución del trabajo “DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO PARA FISIOTERAPIA CON KIENCT”, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y autoría

Sangolquí, 22 de enero de 2016.



Sonia Alejandra Fonseca Factos



## DEDICATORIA

*A Dios por derramar sus bendiciones sobre mí y que con su infinita bondad me ha permitido lograr mis objetivos.*

*A mi ángel de la guarda, mi abuelita (+), que fue un ejemplo de fortaleza, quien me enseñó a salir adelante y no desmayar ante las dificultades.*

*A toda mi familia, especialmente a mis padres y hermana por estar siempre conmigo, brindándome su apoyo incondicional, comprensión y la confianza en cada momento de mi vida. Este logro es tan mío como suyo.*

*Alejandra*

## AGRADECIMIENTO

A Dios, por darme la oportunidad de vivir y por estar conmigo en cada paso que doy, por fortalecer mi corazón e iluminar mi mente y por haber puesto en mi camino a aquellas personas que han sido mi soporte y compañía durante todo el periodo de estudio.

A mis padres Sonia e Iván por su amor incondicional, por los valores inculcados, por sembrar en mí la responsabilidad y deseos de superación, por el esfuerzo que han puesto en mi formación y por ser mi apoyo durante toda mi vida. A mi hermana por su cariño, por tener fé en mí, por su comprensión y por brindarme las palabras que me impulsaron a seguir adelante cuando más lo necesité.

A toda mi familia por creer en mí, por su preocupación, por quererme y apoyarme siempre. De manera especial a mi prima Alexandra por su colaboración en los temas de fisioterapia que fue muy importante para el desarrollo del presente trabajo.

A mis profesores por ejercer con excelencia su labor. Un agradecimiento especial al Ing. Darwin Alulema y al Ing. Flavio Pineda por estar siempre prestos a ayudar brindándome la guía necesaria para desarrollar este proyecto de grado y terminarlo con éxito, por su tiempo y sobre todo por su paciencia.

A todas las personas que de una u otra manera formaron parte de este arduo camino, a mis amigos y compañeros por haber compartido momentos de alegría, tristeza, preocupación, estrés y malas noches dándonos siempre un apoyo mutuo e incondicional.

*Alejandra*

# ÍNDICE GENERAL

<b>CERTIFICADO</b>	<b>ii</b>
<b>DECLARACIÓN DE RESPONSABILIDAD</b>	<b>iii</b>
<b>AUTORIZACIÓN</b>	<b>iv</b>
<b>DEDICATORIA</b>	<b>v</b>
<b>AGRADECIMIENTO</b>	<b>vi</b>
<b>ÍNDICE GENERAL</b>	<b>vii</b>
<b>ÍNDICE DE TABLAS</b>	<b>xi</b>
<b>ÍNDICE DE FIGURAS</b>	<b>xiii</b>
<b>RESUMEN</b>	<b>xvii</b>
<b>ABSTRACT</b>	<b>xviii</b>
<b>1 INTRODUCCIÓN</b>	<b>1</b>
1.1 Antecedentes . . . . .	1
1.2 Justificación e Importancia . . . . .	3
1.3 Alcance del Proyecto . . . . .	5
1.4 Objetivos . . . . .	6
1.4.1 General . . . . .	6

1.4.2	Específicos . . . . .	6
1.5	Estudio del Estado del Arte . . . . .	7
1.5.1	Rehabilitación con el Uso de Kinect . . . . .	14
<b>2</b>	<b>MARCO CONCEPTUAL</b>	<b>23</b>
2.1	Fisioterapia . . . . .	23
2.1.1	Técnicas de Fisioterapia . . . . .	23
2.1.2	Tipos de Fisioterapia . . . . .	25
2.1.3	Ejercicios Terapéuticos . . . . .	26
2.2	Rehabilitación Virtual . . . . .	27
2.2.1	Rehabilitación Virtual Motora . . . . .	30
2.2.2	Rehabilitación Virtual Psicológica . . . . .	31
2.2.3	Impacto Socioeconómico . . . . .	32
2.3	Captura de Movimiento ( <i>MoCap</i> ) . . . . .	33
2.3.1	Tipos de <i>MoCap</i> . . . . .	33
2.4	Kinect . . . . .	35
2.4.1	Componentes . . . . .	35
2.4.2	Funcionamiento . . . . .	38
2.4.3	Campo de Visión del Sensor <i>Kinect</i> . . . . .	39
2.4.4	Software Controlador del Sensor <i>Kinect</i> . . . . .	41
2.4.5	Sistemas de Reconocimientos de Gestos y Posturas . . . . .	44
2.5	Base de Datos . . . . .	46
2.5.1	Base de Datos MySQL . . . . .	47
2.6	Lenguaje PHP . . . . .	48
<b>3</b>	<b>DISEÑO E IMPLEMENTACIÓN</b>	<b>49</b>
3.1	Diseño del Prototipo Auxiliar para Fisioterapia. . . . .	49
3.1.1	Descripción del Sistema. . . . .	49
3.1.2	Requisitos del Prototipo. . . . .	50
3.1.3	Diagrama de Bloques del Prototipo. . . . .	50

3.1.4	Diseño de la Etapa de Captura de Datos. . . . .	51
3.1.5	Diseño de la Etapa de Procesamiento de la Información. . . . .	57
3.1.6	Diseño de la Etapa del Servidor Web y Base de Datos. . . . .	75
3.2	Implementación del Prototipo Auxiliar para Fisioterapia. . . . .	78
3.2.1	Herramientas y Recursos . . . . .	78
3.2.2	Interfaz de Aplicación. . . . .	79
3.2.3	Flujo de Datos del Sensor Kinect. . . . .	81
3.2.4	Reconocimiento de Movimientos. . . . .	88
3.2.5	Reconocimiento de Voz . . . . .	92
3.2.6	Descripción de las Clases y Métodos Implementados. . . . .	94
3.2.7	Diagrama de Clases de la Aplicación. . . . .	103
3.2.8	Implementación de la Aplicación Web. . . . .	105
<b>4</b>	<b>PRUEBAS DE FUNCIONAMIENTO DEL PROTOTIPO</b>	<b>108</b>
4.1	Pruebas de Funcionamiento del Prototipo. . . . .	108
4.1.1	Rango de Reconocimiento del Sensor <i>Kinect</i> . . . . .	108
4.1.2	Funcionamiento de la Aplicación WPF. . . . .	112
4.1.3	Ventana Principal. . . . .	112
4.1.4	Iniciar Sesión y Registrar. . . . .	113
4.1.5	Ejecución de la Rutina de Ejercicios. . . . .	114
4.1.6	Funcionamiento de la Aplicación Web. . . . .	118
4.1.7	Evaluación del Sistema. . . . .	119
4.2	Prueba de Rehabilitación Piloto. . . . .	120
4.2.1	Escenario de Prueba . . . . .	120
4.2.2	Consideraciones de la Prueba . . . . .	121
4.2.3	Pruebas . . . . .	122
4.2.4	Resultados de la Prueba Piloto del Prototipo . . . . .	124
<b>5</b>	<b>CONCLUSIONES Y RECOMENDACIONES</b>	<b>127</b>
5.1	Conclusiones . . . . .	127

5.2 Recomendaciones . . . . . 130

**BIBLIOGRAFÍA** **131**

# ÍNDICE DE TABLAS

1	Componentes Internos de <i>Kinect</i> . . . . .	36
2	Explicación de los Campos de la Arquitectura del SDK de <i>Kinect</i> . . . . .	43
3	Características de las Versiones de <i>Kinect</i> . . . . .	52
4	Comparación entre las Alternativas OpenSource y el SDK de Microsoft. . .	54
5	Nomenclatura de los Tipos de Requisitos de la Aplicación. . . . .	60
6	Requisitos Funcionales del Sistema. . . . .	61
7	Requisitos Funcionales de la Opción " <i>Ejecutar</i> ". . . . .	61
8	Requisitos Funcionales de la Opción " <i>Realizar Ejercicios</i> ". . . . .	62
9	Requisitos No Funcionales de la Aplicación. . . . .	63
10	Modelo de Casos de Uso " <i>Iniciar Aplicación</i> ". . . . .	67
11	Modelo de Casos de Uso " <i>Menú Principal</i> ". . . . .	67
12	Modelo de Casos de Uso " <i>Ayuda</i> ". . . . .	68
13	Modelo de Casos de Uso " <i>Información</i> ". . . . .	68
14	Modelo de Casos de Uso " <i>Ejecutar</i> ". . . . .	69
15	Modelo de Casos de Uso " <i>Recibe y Presenta Información del Paciente y Ejercicio</i> ". . . . .	70
16	Modelo de Casos de Uso " <i>Controla que se Realice Correctamente el Movimiento</i> ".	70
17	Modelo de Casos de Uso " <i>Cerrar Aplicación</i> ". . . . .	71
18	Controles de las Ventanas de la Aplicación. . . . .	79
19	Funciones para Presentar la Imagen de Color. . . . .	84
20	Funciones para Controlar el Flujo de Esqueletos. . . . .	86

21	Clases y Funciones para Controlar el Cronómetro. . . . .	98
22	Métodos de la Clase Cronómetro. . . . .	99
23	Métodos de la Clase ConectarBD. . . . .	102
24	Métodos Utilizados para la Conexión de la BD. . . . .	107
25	Rango de Distancias del Sensor. . . . .	111
26	Verificación de Funcionamiento de la Ventana Principal. . . . .	113
27	Verificación de Funcionamiento de las Interfaces Iniciar Sesión y Registrar. . . . .	114
28	Verificación de Funcionamiento de la Rutina de Ejercicios. . . . .	117
29	Pruebas de Funcionamiento de la Aplicación Web. . . . .	118
30	Datos del Paciente. . . . .	121
31	Datos Obtenidos de la Prueba Piloto. . . . .	125



# ÍNDICE DE FIGURAS

1	Sistemas Comerciales de Rehabilitación. . . . .	8
2	Sistemas Alternativos de Rehabilitación. . . . .	11
3	Robot <i>Turtlebot</i> equipado con el sensor <i>Kinect</i> . <b>Fuente:</b> Katić, Ćosić, Despotović, and Miloradović (n.d.) . . . . .	12
4	Cirujano utilizando la aplicación TedCas. <b>Fuente:</b> Toribios Blázquez (2012)	13
5	Cirujano utilizando la aplicación TedCas. <b>Fuente:</b> Hernández Toala and Herrera Rodríguez (2013) . . . . .	13
6	Esquemas del Sistema VirtualRehab. . . . .	15
7	Entorno de Rehabilitación. <b>Fuente:</b> BRAVO et al. (n.d.) . . . . .	16
8	Estructura del sistema. . . . .	17
9	Diagrama de Bloques del Sistema Mediknect. . . . .	17
10	Estructura del Hardware y Software. <b>Fuente:</b> Moreno et al. (2013) . . . . .	18
11	Estructura General del Framework. <b>Fuente:</b> Moreno et al. (2013) . . . . .	19
12	Interfaz para análisis biomecánico de movimientos. <b>Fuente:</b> Villada and Muñoz (2014) . . . . .	20
13	Esquema Utilizado para el Análisis de Movimientos. . . . .	21
14	Técnicas de Fisioterapia. . . . .	24
15	Movimientos para la Rehabilitación de Articulaciones. <b>Fuente:</b> Tortora and Derrickson (2009) . . . . .	27
16	Continuo de la Virtualidad de Milgram. . . . .	29
17	Componentes del Sensor Kinect. . . . .	36

18	Funcionamiento del Sensor de Profundidad. . . . .	39
19	Campo de Visión de Profundidad. <b>Fuente:</b> Rodriguez Estevez (2013) . . . .	40
20	Campo de Visión Horizontal y Vertical. <b>Fuente:</b> Rodriguez Estevez (2013)	40
21	Arquitectura del SDK de <i>Microsoft</i> <b>Fuente:</b> Rodriguez Estevez (2013) Mi- crosoft (n.d.) . . . . .	43
22	Desplazamiento a la derecha. <b>Fuente:</b> Robledo Zarco (2012) . . . . .	45
23	Conceptos Base de Datos. . . . .	46
24	Descripción del Sistema Auxiliar de Fisioterapia. . . . .	50
25	Sensor Kinect XBOX360. . . . .	53
26	Diagrama de Bloques de la Etapa de Captura de Datos. . . . .	55
27	Estructura del Un Pixel en Formato RGB. . . . .	56
28	Lista de Articulaciones del SDK de <i>Kinect</i> . . . . .	56
29	Plano de Referencia de las Coordenas de las Articulaciones. . . . .	57
30	Diagrama de Bloques de la Etapa de Procesameinto. . . . .	58
31	Diagrama de Flujo de la Aplicación. . . . .	64
32	Diagrama de Casos de Uso. . . . .	65
33	Diagrama de Casos de Uso. . . . .	66
34	Diseño de la Ventana " <i>Principal</i> ". . . . .	72
35	Diseño de la Ventana " <i>Iniciar Sesión</i> ". . . . .	72
36	Diseño de la Ventana " <i>Registrar</i> ". . . . .	73
37	Diseño de la Ventana " <i>Lista de Ejercicios</i> ". . . . .	73
38	Diseño de la Ventana " <i>Opciones</i> ". . . . .	74
39	Diseño de la Ventana " <i>Realizar Ejercicio</i> ". . . . .	74
40	Diseño de la Ventana " <i>Realizar Ejercicio</i> ". . . . .	75
41	Modelo de la Base de Datos . . . . .	76
42	Diseño de la Interfaz de la Aplicación Web. . . . .	77
43	Interfaz de Aplicación. . . . .	80
44	Interfaz de Aplicación de las Ventanas " <i>Realizar Ejercicio</i> " y " <i>Opciones</i> ". . . .	81
45	Sentencias para Iniciar el <i>Kinect</i> . . . . .	82

46	Diagrama de Flujo del Algoritmo utilizado para Iniciar la Transmisión de Datos y Habilitar los Flujos. . . . .	83
47	Sentencias para Habilitar el Flujo de Color del <i>Kinect</i> . . . . .	83
48	Diagrama de Flujo del Algoritmo utilizado para la Imagen de Color. . . . .	84
49	Imagen de Color. . . . .	85
50	Articulaciones Reconocidas por <i>Kinectt</i> . . . . .	85
51	Sentencias para Definir las Coordenadas de los Joints. . . . .	86
52	Diagrama de Flujo del Algoritmo Utilizado para el Control de los Datos de Esqueletos. . . . .	87
53	Imagen Esqueletizada. . . . .	87
54	Vectores de Referencia. . . . .	88
55	Ángulo AVR y AVH para el Ejercicio Hombro. . . . .	89
56	Diagrama de Flujo del Algoritmo Utilizado para Definir el Movimiento a Detectar. . . . .	90
57	Creación del Enumerador. . . . .	90
58	Diagrama de Flujo del Detector de Posturas. . . . .	91
59	Diagrama de Flujo de la Llamada a las Funciones de Detcción de Movimientos. . . . .	92
60	Diagrama de Flujo del Reconocimiento de Voz. . . . .	93
61	Sentencias para la Configuración del Reconocimiento de Voz. . . . .	93
62	Valores que Determinan la Calidad de Captura. . . . .	94
63	Diagrama de Flujo del Algoritmo utilizado para el Verificar el Área de Captura. . . . .	95
64	Diagrama de Flujo del Algoritmo utilizado para el Dibujar Huesos. . . . .	96
65	Diagrama de Flujo del Algoritmo Utilizado para Calcular Ángulo. . . . .	97
66	Setencias para el Método Iniciar el Cronometro. . . . .	99
67	Diagrama de flujo del Algoritmo Utilizado en la Clase Persona. . . . .	100
68	Diagrama de flujo del Algoritmo Utilizado en la Clase Vector3D. . . . .	100
69	Diagrama de flujo del Algoritmo Utilizado en la Clase DatosSesión. . . . .	101
70	Diagrama de flujo del Algoritmo Utilizado en la Clase EnviarInfSesion. . . . .	102
71	Diagrama UML de Clases . . . . .	104

72	Interfaz de Usuario de la Aplicación Web (Ventana index.php) . . . . .	105
73	Interfaz de Usuario de la Aplicación Web(Ventana información.php) . . . . .	106
74	Estructura de la Información de Profundidad . . . . .	109
75	Sentencias para Habilitar la Información de Profundidad . . . . .	110
76	Estructura de la Información de Profundidad . . . . .	110
77	Distancias del Usuario con Respecto al Sensor. . . . .	111
78	Mensajes Informativos de las Interfaces Login y Registrar. . . . .	113
79	Ejecución de la Rutina de Ejercicio. . . . .	115
80	Mensaje Enviado-Bandeja de Entrada . . . . .	116
81	Estructura del Mensaje Enviado . . . . .	116
82	Modelo de Encuesta. . . . .	119
83	Resultados de la Encuesta Aplicada en Base a los Parámetros de Nielsen. . .	120
84	Esquema del Escenario de Prueba . . . . .	121
85	Paciente en la Prueba Piloto. . . . .	123
86	Progreso del Paciente en la Prueba Piloto. . . . .	125

## RESUMEN

El presente proyecto describe el diseño e implementación de un prototipo para fisioterapia con Kinect, que consta de una aplicación desarrollada en Visual Studio 2013 con el lenguaje de programación C# WPF y una aplicación web desarrollada sobre lenguaje PHP. El sistema se encarga de extraer la información de la rutina realizada por el paciente en cada sesión que se utilice el prototipo como auxiliar en el proceso de rehabilitación y almacena en el servidor para que el terapeuta o especialista que supervisa el progreso pueda visualizar en el sitio web. Para el desarrollo del proyecto se ha utilizado un sensor Kinect para Xbox con su respectivo adaptador para poder establecer la conexión con el ordenador, el software oficial (SDK de Microsoft) que permite controlar el sensor y el sistema operativo Windows 8. En el análisis y diseño se contemplan los requisitos funcionales y no funcionales que debe cumplir la aplicación, la selección de las herramientas de desarrollo, características de los ejercicios terapéuticos, los parámetros que son de importancia para el terapeuta y que se van a enviar al servidor, además se realizó el diseño de la base de datos que almacenará la información de cada sesión que realiza el paciente. Tomando en cuenta todas las consideraciones se implementó el software del prototipo y los resultados de las pruebas realizadas demostraron que el comportamiento alcanzado se enmarcan dentro del objetivos propuestos.

### **PALABRAS CLAVE:**

- *SENSOR KINECT*
- C# - WPF
- FISIOTERAPIA
- PHP
- SDK

## ABSTRACT

The present project describe the design and implementation of a prototype for physiotherapy with Kinect, which consists of an application developed in Visual Studio 2013 with the programming language C# WPF and a web application developed on PHP language. The system is responsible for extracting information from the routine performed by the patient in each session in which the prototype is used as an auxiliary in the rehabilitation process and It is stored in a server so that the therapist or specialist that monitors progress can review on the web site. To develop the project has used a Kinect sensor for Xbox with their adapter to connect to the computer, the official software (Microsoft SDK) that allows you to control the sensor and the operating system Windows 8. In the analysis and design is described functional requirements and nonfunctional to be met by the system, the selection of development tools, characteristics of therapeutic exercises, the parameters that are important for the therapist and that be sent to the server, also performed the design of the database that will store the information for each session that made the patient. Taking into account all relevant considerations the prototype software was implemented and the results of tests demonstrated that performance achieved is within the objectives.

### KEY WORDS:

- KINECT SENSOR
- C# - WPF
- PHYSIOTHERAPY
- PHP
- SDK

# CAPÍTULO 1

## INTRODUCCIÓN

### 1.1 Antecedentes

Desde los inicios de la humanidad se ha venido dando importancia a los conceptos de salud y enfermedad, los cuales han tenido una evolución constante a lo largo de la historia en función de los avances médicos y económicos de la sociedad. El estudio realizado en Gallego (2007) sostiene que los restos óseos de generaciones anteriores evidencian la existencia de lesiones, traumatismos, procesos tumorales, infecciones, etc. que fueron tratados con los conocimientos que se tenía en el momento. Para tener una actividad física adecuada, terapias complementarias y un incremento en el bienestar físico, se necesita la ayuda de la fisioterapia que tiene una incidencia social y a la cual la Confederación Mundial de Fisioterapia la define como el conjunto de métodos y técnicas que, mediante la aplicación de medios físicos curan, previenen y adaptan a personas afectadas de algún tipo de disfuncionalidad o a las que desean tener un nivel adecuado de salud manteniendo una movilidad física óptima.

Con el desarrollo de las tecnologías en los últimos años se ha dado un impulso a nuevos medios físicos o equipos que ayuden a realizar las actividades fisioterapéuticas. Hasta el momento, para acceder a estas herramientas el paciente debe asistir a un centro especializado que cuente con especialistas para cada paciente que les ayude a mejorar

su salud mediante la aplicación de sistemas mecánicos y técnicas adecuadas. Con esta modalidad se requiere de una cita realizada previamente o tiempo de espera para ser atendido, en muchos casos se debe cancelar un valor por cada sesión de rehabilitación, que no está al alcance de toda la población que requiere este tipo de servicio de salud. Además el traslado del paciente hacia el centro de rehabilitación se dificulta al no encontrarse en los alrededores o cerca del sector.

Debido al incremento del número de personas que requieren el servicio de fisioterapia para mejorar su calidad de vida, se ha visto la manera de reducir los costos y el tiempo del tratamiento para que esté al alcance de todos, es por eso que aparece el concepto de sistemas de bajo costo. Este sistema consiste en la aplicación de realidad virtual, que en este caso se utiliza para conseguir tener rehabilitación virtual que en SANCHIS CANO (2013) se define como la unión de dos campos: la rehabilitación y la realidad virtual. Con este nuevo esquema de rehabilitación no es necesario que el paciente acuda a un hospital para realizar sus sesiones de fisioterapia, ya que con la ayuda de sensores de movimientos, que muchas veces tienen las consolas de videojuegos, y software desarrollado específicamente para estos fines se puede realizar la terapia en la comodidad del hogar.

En las últimas décadas han salido al mercado una variedad de consolas de videojuego, las cuales fueron utilizadas por las personas en un ambiente doméstico con el único fin de divertirse y compartir con amigos, familiares y conocidos; inicialmente, estas eran manejadas principalmente por una palanca para realizar movimientos dentro del juego. Cuando la tecnología fue avanzando se pasó al uso de un control, el cual se conectaba a la consola por medio de un cable, desembocando en la actualidad al uso de un sensor inalámbrico de movimiento (Román Barrera, Sanabria Daza, Tengonó Ramírez, et al., 2014), denominado Kinect que fue desarrollado por Microsoft, viene incluido en la videoconsola XBOX 360, y que permite que la interacción con el mundo virtual sea de una forma más natural. Este sensor ha sido utilizado en múltiples aplicaciones en las que interviene el cuerpo humano tales como, entrenamiento de



karate que se detalla en Rodríguez Estevez (2013), estudios para obtener la captura de objetos que se realizó en Nuño Simón (2012), juegos educativos, etc.

Con la implementación las herramientas que contribuyen a lograr la rehabilitación virtual (sensor de movimiento y software) se pretende conseguir un apoyo para la labor que el terapeuta desarrolla y brindar las comodidades que los pacientes requieren para mejorar su condición física y su salud en general.

## **1.2 Justificación e Importancia**

Según el Informe Mundial sobre la Discapacidad: Se estima que más de mil millones de personas viven con algún tipo de discapacidad; o sea, alrededor del 15% de la población mundial, según estimaciones de la población mundial del año 2010 (CONADIS, 2012).

En CONADIS (2012) también se tiene que la Encuesta Mundial de Salud refleja que del total estimado de personas con discapacidad 110 millones (2,2%), tienen dificultades muy significativas de funcionamiento, mientras que a Carga Mundial de Morbilidad cifra en 190 millones (3,8%), las personas con discapacidad grave (el equivalente a la discapacidad asociada a afecciones como la tetraplejía, depresión grave o ceguera).

Según los datos arrojados del último Censo de Población y Vivienda realizado en el año 2010, en el Ecuador el 5,6 % de la población tiene alguna discapacidad, en el caso de la discapacidad física puede ser permanente o temporal. Por otro lado, el estudio "Ecuador: la discapacidad en cifras" realizado por el CONADIS en el año 2005 establece que el 12.14% de la población ecuatoriana tiene algún tipo de discapacidad; y, el estudio bio-psico-social Misión Solidaria Manuela Espejo, da cuenta de 294.803 personas con discapacidad (CONADIS, 2012).

En base a las estadísticas descritas se puede determinar que la porción de la población,

de todas las edades y género, que presenta alguna discapacidad o impedimento físico permanente o temporal debe tener acceso a algún tipo de rehabilitación física, y de esa manera cumplir con lo que se describe en CONADIS (2012), que en el Ecuador, el Gobierno de la Revolución Ciudadana ha asumido el reto de construir una "Patria para todos y todas", una sociedad más justa y solidaria que promueva la inclusión social de todos los sectores, especialmente de aquellos grupos que requieren atención prioritaria como es el caso de las personas con discapacidad. Esta consideración consta también el Código del Buen Vivir, el mismo que hay que tomar muy en cuenta al momento de ejecutar un proyecto.

Para mejorar la condición de salud del paciente, éste debe acudir a varias sesiones de fisioterapia. Considerando que una rehabilitación tradicional tienen un costo que oscila al rededor de 20 dólares por sesión y que se debe acudir constantemente a un centro especializado que cuente con los equipos necesarios hasta completar el tratamiento, hay mucha gente que no completa sus ejercicio por el tiempo y el costo que representa, esto constituye un riesgo para su salud, que en un futuro se traduce en más problemas y gastos. En este tipo de rehabilitación un terapeuta está vigilando que su paciente realice los ejercicios de forma correcta, de igual forma el terapeuta decide si es necesario aumentar la intensidad de la rutina o bajarla según los resultados de su paciente.

Hoy en día se tiene la alternativa y la tecnología necesaria para desarrollar un prototipo auxiliar y a la vez comercial que mediante la rehabilitación virtual permita realizar sesiones de fisioterapia desde la comodidad del hogar. La importancia que tiene ésta investigación es que al utilizar sistemas que reduzcan en un 40 % los costos actuales de la sesión, que se consigue al aplicar la realidad virtual, se puede solventar las dificultades que tienen muchas personas de trasladarse al centro de fisioterapia y de no poder acceder a un seguro de salud, el 60,2% de mujeres y el 57,9 % de hombres según los resultados presentes en INEC (2010) basado en el CENSO realizado en el año 2010, que cubra los gastos de las sesiones de fisioterapia que permiten lograr una mejor calidad de vida. Logrando así un impacto socioeconómico positivo, ya que el

nuevo mecanismo es amigable con el usuario y puede estar al alcance de todo el que lo requiera, permitiendo realizar ejercicios destinados a rehabilitar problemas específicos como problemas de equilibrio y la falta de coordinación, los trastornos del movimiento y de la postura y los déficits motores, entre otros. Además, se puede realizar la rutina desde el hogar y con el monitoreo del terapeuta.

### **1.3 Alcance del Proyecto**

El proyecto se basa en la utilización de un sensor KINECT de Microsoft y presenta como producto final un prototipo para procesos de rehabilitación física básica en las que intervienen movimientos de las extremidades superiores e inferiores del cuerpo humano, las cuales son identificadas por el sensor. Las rutinas que se puede trabajar con el prototipo constan de ejercicios repetitivos que permitan mejoras en la marcha, la resistencia, el equilibrio, coordinación y fortalecimiento muscular. Las sesiones pueden ser realizadas tanto en el centro de tratamiento como en la propia casa del paciente, que es el actor principal del proceso, ya que esta herramienta permite que los ejercicios sean monitorizados por el especialista para evaluar la evolución, constituyendo un apoyo a la labor que el terapeuta desarrolla.

La aplicación consiste en una interfaz gráfica amigable con el usuario, realizada mediante lenguaje de programación orientada a objetos, en la cual el paciente puede visualizar el nombre del ejercicio, el patrón de ejecución, el número de repeticiones que debe realizar y las que se van contabilizando conforme avance la rutina de rehabilitación. Esta aplicación también permite que desde la aplicación del paciente se conecte con un servidor, el cual realizará la comprobación de los movimientos y almacenará el progreso de las sesiones de rehabilitación. La comprobación se realiza mediante algoritmos de reconocimiento de posturas que permiten realizar una comparación de los movimientos que ejecuta el paciente con respecto a un patrón especificado por el terapeuta, reduciendo así las limitaciones de tiempo y espacio.

## **1.4 Objetivos**

### **1.4.1 General**

- Diseñar e implementar un prototipo para asistir a pacientes en proceso de fisioterapia utilizando un sensor de movimiento Kinect.

### **1.4.2 Específicos**

- Realizar un estudio acerca del sensor Kinect; sus características, modos de operación y métodos de identificación de imágenes.
- Determinar las aplicaciones, características y limitaciones que presenta el dispositivo Kinect.
- Analizar los parámetros de comportamiento de las variables que se obtienen del sensor Kinect necesarias para el desarrollo del proyecto.
- Desarrollar una breve revisión de conceptos de fisioterapia para establecer los parámetros necesarios que se considera al momento de ejecutar una rutina de ejercicios.
- Diseñar los bloques funcionales que formarán parte de la aplicación.
- Evaluar el funcionamiento del sistema mediante pruebas en diferentes situaciones.

## 1.5 Estudio del Estado del Arte

Actualmente el diseño, la ingeniería y la medicina han encontrado una brecha muy estrecha en donde la participación de estas disciplinas ha permitido el desarrollo de proyectos que buscan como base tratar temas de la salud. Dentro del campo de la rehabilitación también se ha visto una notable evolución, ya que se puede encontrar muchos equipos comerciales que ayudan en la labor que el fisioterapeuta realiza, razón por la cual se tienen laboratorios que cuentan con las últimas tecnologías, que son aprovechadas para desarrollar programas de fisioterapia. Según investigaciones previas, considerando aquellas realizadas desde el año 2012, de proyectos de diseño relacionadas con temas de salud en rehabilitación, terapias de rodilla, aplicaciones interactivas, diseño de productos, se ha encontrado los siguientes dispositivos o equipos, como resultado de diferentes estudios alrededor del mundo, capaces de ayudar en la rehabilitación de pacientes con cierto grado de discapacidad.

- **Lumo Back (LUMO, 2012).**- es un sensor, desarrollado por Monisha Perakash, Charles Wang, Andrew Chang y Andreas Hauenstein en el año 2012, que proporciona una suave vibración para recordarle al usuario que se encuentra con una postura incorrecta, ya sea que se encuentra de pie o sentado. Se lleva en la espalda baja y diseñado para ser delgado, elegante y tan cómodo que apenas se pueda sentir cuando se lo tenga puesto. El sensor se conecta de forma inalámbrica a una aplicación móvil gratuita que rastrea las actividades diarias (ver Figura 1(a)).
- **Nike+ fuelband(Nike, 2012).**- Es una pulsera creada y distribuida por Nike (ver Figura 1(b)), capaz de realizar un seguimiento a las actividades diarias, según el movimiento realizado por el usuario los indicadores se activan, de rojo a verde, dependiendo de la intensidad de la actividad. También se puede visualizar las estadísticas del progreso por medio de una aplicación móvil compatible con iOS. La innovación de este dispositivo está en la forma de crear motivación en las per-

sonas para que se preocupen de realizar actividad física, dejando de ser sedentarias y trabajando para tener una vida saludable al convertirse en gente activa.

- **NedSVE/IBV- Aplicación para la Valoración y Rehabilitación del Equilibrio**(Instituto de Biomecánica de Valencia, n.d.)- es una aplicación desarrollada por el Instituto de Biomecánica de Valencia, que es utilizada para para el análisis, reeducación y seguimiento de trastornos del equilibrio con resultados gráficos fácilmente interpretables. La función del equilibrio se considera el resultado de un complejo sensorial y motor, cuya finalidad es el mantenimiento de la postura. La aplicación permite combinar pruebas de posturografía estática y dinámicas basadas en el análisis de marcha, límites de estabilidad y seguimiento con el centro de presiones. El esquema de la aplicación *NedSVE/IBV* se puede apreciar en la Figura 1(c).



(a) LumoBack y su aplicación móvil. (b) Manilla y Aplicación de Nike. (c) Valoración y Rehabilitación del Equilibrio.

**Fuente:** LUMO (2012)

**Fuente:** Nike (2012)

**Fuente:** Instituto de Biomecánica de Valencia (n.d.)

Figura 1: Sistemas Comerciales de Rehabilitación.

Por otro lado, gracias a la evolución que la realidad virtual ha tenido o está experimentando en los últimos años y a las ventajas que ofrece, se han venido desarrollando proyectos dentro del campo de la rehabilitación que involucran realidad virtual y sistemas alternativos a los tradicionales, los cuales se encuentran dentro de los video-

juegos, ya que por las características que presentan sus sistemas periféricos (mandos) o sus consolas se convierten en una herramienta que sirve para la interacción de los usuarios con los sistemas. A continuación se explica el funcionamiento de tres alternativas de las que se podrían disponer para su aplicación en rehabilitación.

- ***Wii Remote de Nintendo***

El *Wii Remote* es el mando principal de la consola *Wii* de Nintendo y sus sucesoras, cuya característica principal es la comunicación inalámbrica que tiene con la consola mediante el protocolo de comunicación Bluetooth(IEEE 802.15.1-2005). Por ello, el *Wii Remote* puede funcionar como un apuntador de objetos en la pantalla y sobre todo, un detector de movimiento en 3D, interactuando mediante gestos físicos y presión sobre los botones. El reconocimiento del usuario se realiza mediante la combinación de los sensores LEDs infrarrojos que posee en la barra de sensores y la información que proviene del acelerómetro. El aspecto físico del controlador se puede apreciar en la Figura 2(a).

Como herramienta para rehabilitación, la *Wii* ha sido utilizada en las áreas de rehabilitación de algunos hospitales alrededor del mundo y se le ha otorgado un nombre específico a este tipo de terapia **Wii-Habilitation**. Es utilizada como actividades complementarias bajo la supervisión de un profesional y no como un tratamiento que el usuario realiza de forma independiente.

- ***Playstation Move***

Es un sistema que controla la consola *Playstation*, es un sistema basado en la tecnología de detección de movimientos, consta de tres partes principales que son el *Motion Controller*, *Navigation Controller* y *Playstation Eye* (Figura 2(b)), su comunicación es de forma inalámbrica a través de Bluetooth 2.0 y cuentan con una batería de ion de litio.

***Motion Controller***.- es el amndo principal de *Playstation Move*, tiene forma rectangular con una esfera en la parte superior que cambia de color y se

ilumina de tal manera que se logre diferenciar en la escena. Sirve como marcador activo para lograr conocer su ubicación en el plano en el que se desarrolla el movimiento y gracias a su forma redonda se puede determinar la intensidad o tamaño de la luz para calcular la posición en la que se encuentra el usuario. Además, este controlador cuenta con sensores en su interior como acelerómetros y otro tipo de sensores que permiten controlar el movimiento, la rotación, calibrar la orientación y corregir algún tipo de error que puede generar debido al seguimiento insuficiente de otro componente de *Playstation Move*.

***Navigation Controller***.- es originalmente conocido como subcontrolador de movimiento, ya que replica la funcionalidad principal de la parte izquierda de un mando inalámbrico estándar de Playstation.

***Playstation Eye***.- Se trata de un dispositivo *WebCam* y mediante el cual se reconocen los controles del PlayStation Move, ya que detecta el color del control y lee los movimientos de éste, los cuales luego son representados en el juego. La cámara tiene una resolución de 640x480 píxeles a una frecuencia de 60 Hz.

Debido a las características de sus componentes se puede decir que es un dispositivo adecuado para ser usado en ejercicios de rehabilitación por su detección precisa de movimientos.





(a) Wii Remote

(b) Playstation Move

**Fuente:** Nintendo (n.d.)**Fuente:** Playstation (n.d.)

(c) Kinect.

**Fuente:** XBOX (n.d.)

Figura 2: Sistemas Alternativos de Rehabilitación.

- ***Kinect***

El kinect es una barra horizontal con una base en la que se encuentra un eje motor, dispone de una cámara RGB, un sensor de profundidad y un arreglo de micrófonos (Figura 2(c)). Estos componentes hacen posible que los usuarios puedan controlar e interactuar con la Xbox 360 sin necesidad de usar mandos tradicionales o una interfaz táctil, solo usando gestos, comandos de voz y el propio cuerpo.

Debido al libre acceso a los controladores el *Kinect* tiene un amplio campo de aplicación en diferentes áreas tales como la domótica, la medicina (rehabilitación y cirugía), el automovilismo, la robótica, el modelado de objetos y lugares 3D, educación, arte etc. En Hernández Toala and Herrera Rodríguez (2013) se pre-

sentan las siguientes aplicaciones en la que se utiliza el sensor *Kinect*.

**Turtlebot.** es un robot asistente ideal para llevar o traer cosas, que fue diseñado gracias a las ventajas que presenta la tecnología del sensor *Kinect*, es económico, que está equipado con componentes básicos, como un computador portátil, un robot *iRobot Create* y en especial un sensor 3D *Microsoft Kinect*. Este sensor permite captar el ambiente circundante y que recorra eficazmente, las ordenes se ingresa en el computador portátil, que tiene un sistema de software abierto, lo cual permite la reprogramación y presentar o diseñar mejoras en el mismo. en la Figura 3 se puede apreciar la estructura del robot.



Figura 3: Robot *Turtlebot* equipado con el sensor *Kinect*.

**Fuente:** Katić et al. (n.d.)

**Nuevo Asistente en el Quirófano.** Para minimizar las posibilidad de que los pacientes contraigan infecciones en el quirófano, dos investigadores españoles crearon TedCas, una aplicación basada en Kinect, que permite a los médicos que intervienen en la cirugía buscar en el computador toda la información del expediente del paciente, sin necesidad de tocar nada (Figura 5).



Figura 4: Cirujano utilizando la aplicación TedCas.

**Fuente:** Toribios Blázquez (2012)

**Espejo Virtual para Tienda de Ropa.** El 'Magic Mirror' de Intel es una experiencia de compra virtual que utiliza a un avatar del cliente vestido con las prendas que desea probarse. Una pantalla muestra al avatar en 3D, el cual sigue los movimientos de la persona en tiempo real y cambia las dimensiones del cuerpo usando gestos. Por otro lado, la tienda Bloomingdale está utilizando Kinect, en su almacén de Los Ángeles, para hacer un mapa 3D del cuerpo y así producir 'el perfecto par de jeans' para cada uno de sus clientes.



Figura 5: Cirujano utilizando la aplicación TedCas.

**Fuente:** Hernández Toala and Herrera Rodríguez (2013)

**Estudio de Movimientos en Glaciales.** El equipo requerido para realizar mediciones en 3D a superficies como la de un glaciar puede costar entre 10.000 y 200.000 dólares. Sin embargo, el investigador Marco Tedesco, de la Universidad City College de Nueva York, compró un Kinect de 120 dólares que acopló a un helicóptero a control remoto para tomar imágenes de las lagunas que se forma sobre los glaciares, cuando estos se derriten, cuya agua dreña por grietas y acelera el camino de estos hacia el mar. Así pudo anticipar este fenómeno.

**Anuncios Publicitarios Inteligentes.** Un programa que se desarrolla en los laboratorios de Intel, llamado Advertising Framework, reconoce la edad y sexo de las personas mediante las cámaras y el sensor de Kinect, con el fin de mostrarles de forma inteligente el anuncio publicitario que más se oriente hacia sus intereses. La idea es ubicarlos en lugares como centros comerciales, hoteles, aeropuertos y zonas comerciales, para que la publicidad sea dirigida a quien realmente corresponda.

### **1.5.1 Rehabilitación con el Uso de Kinect**

La tecnología *Kinect* como un método de rehabilitación física, se ha consolidado en los últimos años convirtiéndose en una gran alternativa al tratamiento convencional debido a que facilita el acceso, la frecuencia y la duración de las rutinas de fisioterapia y motivando al paciente gracias a su aspecto lúdico e interactivo. Se ha podido notar que el paciente se encuentra mucho más animado y con una buena predisposición para seguir los entrenamientos, que resultan mucho más sociables y divertidos (AMarcos(CosasdeSalud), n.d.). Los sistemas de rehabilitación con el uso de *Kinect* que se presentan, son el resultado de la revisión bibliográfica de trabajos desarrollados en la línea de investigación en la que se basa el presente proyecto, considerando únicamente los elaborados desde el año 2013.

- **VirtualRehab (VirtualRehab, n.d.)**- es un sistema desarrollado en España por la empresa VirtualWare. Consiste en ofrecer a sus pacientes un entorno virtual y divertido para desarrollar las terapias.

El sistema permite, utilizando tecnología *Kinect*, un acceso y control online, de tal manera que el fisioterapeuta tenga la opción de editar las terapias y realizar un análisis del progreso del paciente, quien a su vez puede realizar las rutinas desde el hogar si no tiene la posibilidad de acceder a un centro de rehabilitación por motivos de salud o de distancia, esta breve descripción se puede apreciar de forma gráfica en la Figura 6 (a) correspondiente al esquema general del sistema.

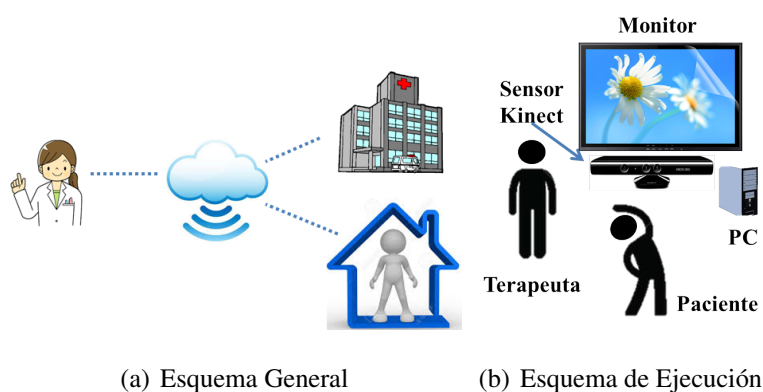


Figura 6: Esquemas del Sistema VirtualRehab.

La Figura 6 (b) muestra un esquemático del entorno de la ejecución de la rutina y consta de un monitor que permite visualizar la interfaz gráfica, un PC que debe contar con sistema operativo Windows y el sensor *Kinect* que realiza la captura de movimientos.

VirtualRehab se basa en la rehabilitación por medio de videojuegos y por ser un sistema comercial es muy completo y ofrece un sin número de opciones y alternativas para ejecutar la sesión. Está dirigido a tratar diferentes patologías obteniendo un gran éxito en la recuperación de las condiciones de movilidad de

los pacientes.

- **Plataforma accesible en el marco de la rehabilitación físico-cognitiva (BRAVO et al., n.d.)** se trata de una plataforma que permite a los usuarios realizar los ejercicios mediante el movimiento del cuerpo, dispone de dos secciones o tipos de rehabilitación física y cognitiva, las mismas que pueden ser configuradas de acuerdo a la necesidad. El funcionamiento y la estructura de la interfaz es muy similar a la de un videojuego (Figura 7), en la que intervienen objetos que sirven de indicador del movimiento que el usuario debe realizar, cada vez que el objeto es alcanzado y se pinta de verde se considera una ejecución correcta, si se pinta de rojo es un movimiento erróneo.



Figura 7: Entorno de Rehabilitación.

**Fuente:** BRAVO et al. (n.d.)

Las librerías empleadas en el desarrollo son las que ofrece OpenNI que es un kit de desarrollo de software de código abierto. Mientras que el sistema operativo utilizado para la implementación de la plataforma fue Windows 7 y básicamente se compone de tres etapas (Figura 8): en la primera intervienen los movimientos del usuario los mismos que son capturados por el sensor *Kinect* y procesados en un PC que permite visualizar mediante un monitor las interfaces.



Figura 8: Estructura del sistema.

Con la implementación de la solución descrita se logró tener una plataforma lo suficientemente robusta y flexible como para incorporar y ofrecer ejercicios para un gran número de padecimientos.

- **MEDIKNECT (SANCHIS CANO, 2013).**- es el resultado de un proyecto realizado en la Universidad Politécnica de Valencia, consiste en una aplicación que utilizando tecnología de captura de movimiento y por medio del sensor *Kinect* permite realizar una rutina de rehabilitación de diferentes partes del cuerpo con ejercicios que dispone la aplicación, los mismos que son grabados y guardados por el médico o fisioterapeuta, el diagrama de bloques que describe el funcionamiento del sistema se tiene en la Figura 9.

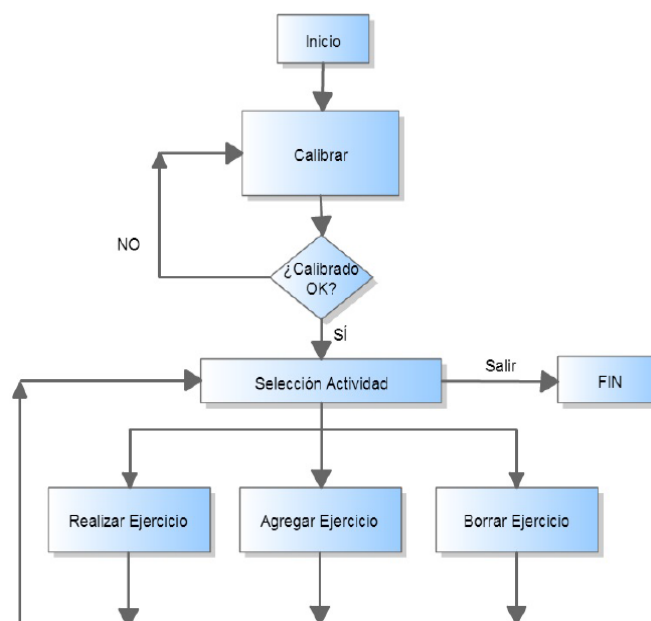


Figura 9: Diagrama de Bloques del Sistema Mediknect.

Con este trabajo se ha obtenido un sistema completamente funcional, pero al no realizar pruebas con pacientes reales no se evalúa la efectividad de las rutinas con ayuda de Mediknect.

- ***Un Framework para la Rehabilitación Física en Miembros Superiores con Realidad Virtual (Moreno et al., 2013).***- el sistema consiste en la creación de hardware y software que permite ejecutar una serie de videojuegos que propicien actividades físicas específicas que sirvan para rehabilitar las extremidades superiores de un paciente. La arquitectura del hardware consta de una estación de trabajo con una PC, el sensor *Kinect* un monitor (Figura 10 (a)). Mientras que el software consta de una interfaz que permite acceder a las opciones establecidas como videojuegos, gestión del paciente o calibración del sistema(Figura 10 (b)).

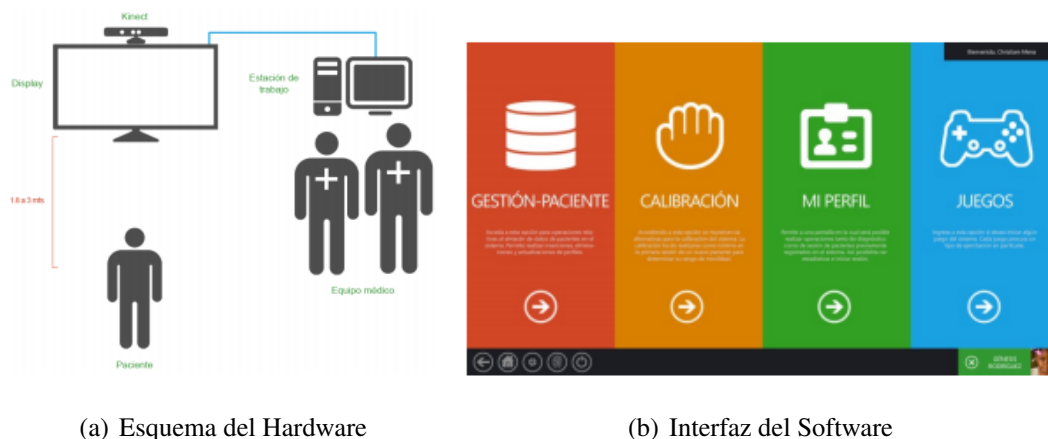


Figura 10: Estructura del Hardware y Software.

**Fuente:** Moreno et al. (2013)

El sistema también cuenta con un guante virtual que permite ejercitar los dedos y por ende la motricidad fina. Utiliza también un controlador en el que se encuentran las interfaces del framework, controlando de esa manera los demás módulos, y un motor que permite agilizar y facilitar la creación de los videojuegos que en este caso se desarrollo un esquema de escena basado en XML. Por otro lado, se



tiene también la base de datos, las estadísticas de la sesión y el sensor *Kinect*, las dos primeras permiten gestionar y administrar las sesiones de rehabilitación y el segundo realiza la captura de datos. Esta descripción se puede apreciar en la estructura general del Framework presente en la Figura 11.

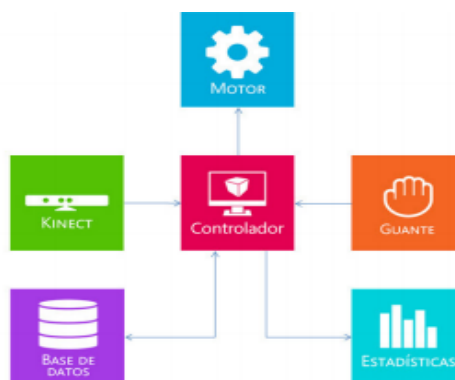


Figura 11: Estructura General del Framework.

**Fuente:** Moreno et al. (2013)

En este estudio se logra evaluar, después de sesiones de prueba, el funcionamiento del sistema en base a las heurísticas de Nielsen que reflejan la conformidad del paciente y aspectos que se debe mejorar en futuras versiones. En esta versión del trabajo se explica también que en una tercera fase se proyecta el evaluación del sistema en pacientes con el respectivo análisis del progreso.

- ***Desarrollo de un software de análisis biomecánico a través de datos de captura de movimiento usando el sensor KINECT para rehabilitación asistida con video juegos (Villada & Muñoz, 2014).***- Es una aplicación desarrollada en Matlab con las librerías de *Kinect* necesarias para realizar el análisis biomecánico de los movimientos realizados por el usuario en la ejecución de un videojuego, se basa principalmente en el análisis de los ángulos de Euler y la posición de los joints medida en centímetros. Básicamente el software consiste en capturar los movimientos en un archivo MoCap que se generan al ejecutar un videojuego diseñado para la rehabilitación, el mismo que puede ser cargado en la interfaz

para poder visualizar la gráfica que relaciona los joints con los del usuario en un espacio tridimensional. La arquitectura que tiene el sistema es similar a la que se evidencia en la Figura 8. La estructura de la interfaz de usuario del software desarrollado que se obtuvo es la que se presenta en la Figura 12.

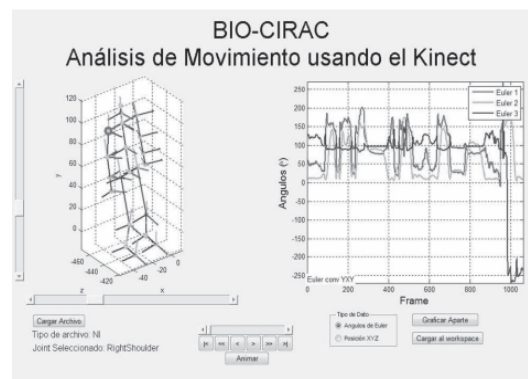


Figura 12: Interfaz para análisis biomecánico de movimientos.

**Fuente:** Villada and Muñoz (2014)

Las pruebas fueron realizadas en un centro de rehabilitación con seis pacientes que padecen de un síndrome que afecta a las motoneuronas denominado síndrome piramidal. Los pacientes se relacionaron con videojuegos diseñados para la rehabilitación del hombro. Los resultados del análisis con ayuda del software desarrollado reportaron que se tiene una mejoría de un 18% en rangos de movimiento en cuatro meses de ejecución de la rutina utilizando un videojuego.

- ***Análisis del movimiento de las extremidades superiores aplicado a la rehabilitación física de una persona usando técnicas de visión artificial (Velarde, Perugachi, Romero, Sappa, & Vintimilla, 2015).***- es un trabajo publicado en agosto de 2015 en la Revista Tecnológica ESPOL, consiste en un sistema para realizar un análisis de los movimientos de las extremidades superiores (hombro y clavícula) de una persona. Los resultados se pueden visualizar de forma gráfica en una interfaz que presenta curvas comparativas en base a los ángulos, calculados con las coordenadas de los joints proporcionadas por el sensor *Kinect*, que

se tiene en los movimientos de abducción, aducción, flexión y extensión de las articulaciones.

Como en los anteriores sistemas el usuario debe realizar los movimientos que visualiza en la interfaz, los cuales son identificados por el sensor *Kinect*, la información es procesada y presentada con ayuda de un computador. Los resultados y las gráficas generadas son interpretadas por el especialista para determinar el progreso de la rehabilitación. El esquema del sistema se tiene en la Figura 13.

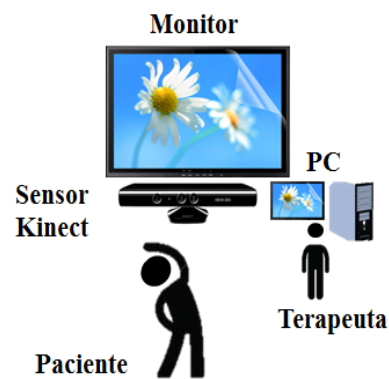


Figura 13: Esquema Utilizado para el Análisis de Movimientos.

Lo que este estudio logró es obtener un sistema que permita analizar los movimientos ejecutados por un paciente en proceso de rehabilitación mediante gráficas generadas en base a la medición de ángulos articulares calculados con la información procedente del sensor. Con ayuda de los datos conseguidos en cada sesión es posible realizar un análisis comparativo, que por proporcionar valores de la medición de ángulos y gráficas complejas son únicamente interpretados por un especialista. Mientras que el usuario se encuentra limitado a realizar los movimientos indicados por el terapeuta.

Cada uno de los proyectos descritos en esta sección son realmente importantes para el desarrollo de este trabajo, ya que describen las formas en las que se está tratando las terapias físicas con las nuevas tecnologías. Además ofrecen un amplio panorama para

entender y permitir considerar otras posibilidades de rehabilitaciones físicas a las que tengan acceso todas las personas que requieran continuamente éstas sesiones.

En el presente trabajo se propone la elaboración de un prototipo auxiliar para fisioterapia, el mismo que consta de una interfaz, diferente a los entornos de videojuegos que se utilizan en la mayor parte de proyectos realizados con el sensor *Kinect*, que proporciona al usuario una serie de movimientos para rehabilitar tres tipos de articulaciones: hombro, cadera y codo. La arquitectura del hardware que requiere el sistema es similar a la de los mencionados en esta sección, en la que involucra el PC, un monitor y el sensor *Kinect* para la ejecución, captura y procesamiento de los movimientos. La diferencia radica en que el paciente tienen una participación importante, debido a que el sistema es amigable y de fácil entendimiento para el usuario, por lo que no es necesaria la intervención del terapeuta para el manejo de la interfaz.

Cabe mencionar que el prototipo no se centra en el análisis biomecánico de los movimientos para determinar el progreso de la rehabilitación, sino que utiliza ciertos fundamentos, como el cálculo de ángulos entre articulaciones, para realizar una comparación y conocer si las posturas especificadas están correctamente ejecutadas, determinando el progreso en base al número de repeticiones y a la duración de la sesión.

Como aporte, el trabajo permite el manejo de ciertas ventanas de la interfaz mediante comandos de voz y posibilita el acceso al especialista a una pagina web, para que pueda visualizar el cumplimiento de la rutina, los resultados y progreso de la rehabilitación.

# CAPÍTULO 2

## MARCO CONCEPTUAL

### 2.1 Fisioterapia

La fisioterapia una alternativa terapéutica sin la utilización de fármacos, tiene como principal objetivo mejorar las condiciones de vida de personas con problemas de movilidad, ayuda a recuperar las funciones, mitigar el dolor y prevenir más lesiones, con el uso de métodos como los ejercicios, estiramientos, estimulación eléctrica y masajes. Además utiliza herramientas especiales como muletas, prótesis estimulación de respuesta por ordenador, etc.

La fisioterapia es muy utilizada en muchos procesos clínicos siempre que en ellos esté indicada la aplicación de terapia física y bajo la supervisión de un profesional. En la rehabilitación es muy utilizada para ejercitarse después de una lesión o una intervención quirúrgica, para tratar problemas de lumbalgia, parálisis cerebral y para aliviar dolores en las articulaciones.

#### 2.1.1 Técnicas de Fisioterapia

La fisioterapia incluye el tratamiento de las patologías del cuerpo humano a través de medios físicos como la temperatura, la electricidad, el movimiento, el ejercicio terapéutico, el agua, el masaje, etc. Todas estas opciones hacen posible que existan

un sin fin de posibilidades terapéuticas al alcance del fisioterapeuta, por lo que se han clasificado en las categorías que se observa en la Figura 14.

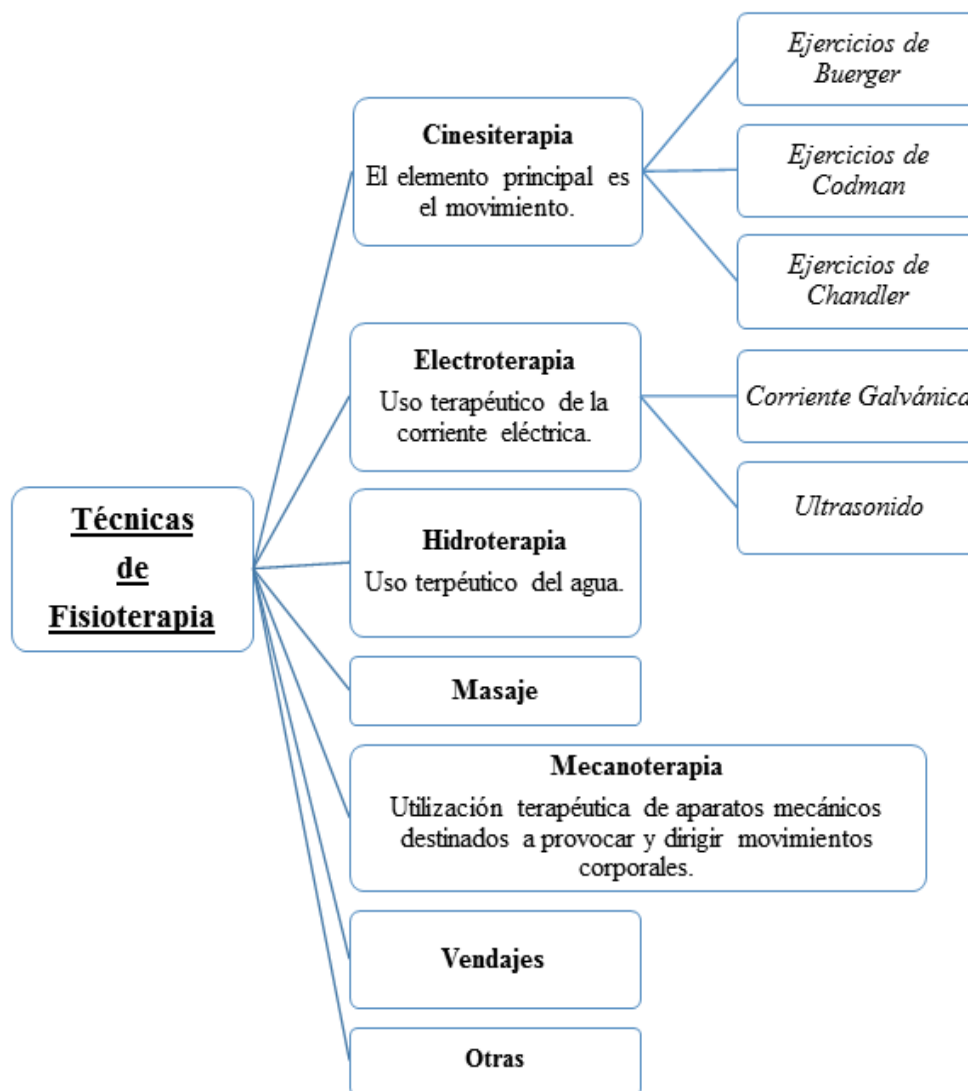


Figura 14: Técnicas de Fisioterapia.

Dentro de la **Cinesiterapia** se observan tres tipos de ejercicio que dependiendo de la situación se aplica, es así que los *Ejercicios de Buerger* consisten en cambios posturales y se utiliza en la rehabilitación de pacientes con patología circulatoria periférica. También están los *Ejercicios de Codman y Chandler*, ambos sirven para tratar problemas o limitaciones de la articulación del hombro con la diferencia que el segundo aporta mayor comodidad y seguridad al paciente.

En la categoría de *Electroterapia* se tienen todos los tipos de corrientes que ayudan en los tratamientos. Por ejemplo, la *Corriente Galvánica* es un tipo de corriente de bajo voltaje que se mantiene constante durante todo el período de tiempo que dura la sesión de fisioterapia. El *Ultrasonido* también forma parte de ésta categoría y son ondas sonoras que generan vibraciones mecánicas, provocando compresiones y dilataciones periódicas en la materia, propagándose a través de ella.

Hay que destacar que la fisioterapia no se limita a un conjunto de técnicas aplicadas sobre el paciente, sino que los fisioterapeutas buscan alternativas y tratan a las personas dependiendo de la situación bio-psico-social(Online, n.d.).

### 2.1.2 Tipos de Fisioterapia

Debido a que las sesiones de fisioterapia están basadas y se adaptan a la afección del paciente, existen varios tipos de fisioterapia dependiendo del área afectada, entre las más conocidas están las siguientes.

- **Musculoesquelética.-** permite tratar la flexibilidad y la fuerza mediante movimientos y masajes en las articulaciones.
- **Cardiovascular.-** permite el tratamiento de problemas cardíacos y pulmonares mediante ejercicios que aumenten los movimientos del paciente, eliminando la mucosidad de los pulmones y permitiendo una correcta ventilación de los mismos para tener una correcta respiración.
- **Neurológica.-** se aplica en casos en los que se tiene trastornos cerebrales y del sistema nervioso como enfermedad de Parkinson o de Alzheimer, permitiendo trabajar y recuperar el equilibrio, la coordinación y las funciones motoras.

En cada uno de estos tres tipos de fisioterapia que se han mencionado el fisioterapeuta analiza la situación y aplica las técnicas adecuadas y que sean beneficiosas para la recuperación del paciente.

El prototipo que se diseña con tecnología *Kinect* sirve como elemento auxiliar en tratamientos en los que interviene una rehabilitación con ejercicios repetitivos, ya que a través del método de captura de movimiento ofrece al paciente rutinas de una manera entretenida y amigable para cumplir completamente el programa de fisioterapia que el médico tratante haya recomendado. Al utilizar un sistema alternativo en la rehabilitación, como es el sensor *Kinect*, se adentra en el campo de la ***Rehabilitación Virtual***

### 2.1.3 Ejercicios Terapéuticos

Los ejercicios terapéuticos consisten en los métodos en los que intervienen movimientos utilizados para ejecutar sesiones de rehabilitación de zonas musculares o articulares afectadas por algún tipo de padecimiento. Dependiendo de la mayor o menor participación del paciente en la ejecución de los movimientos y la ayuda a resistencia aplicada manual o por mecanismos externos diversos, se clasifican en dos grupos generales que son *Ejercicios Pasivos* y *Ejercicios Activos*.

- **Ejercicios Pasivos.-** son ejercicios en los que no se requiere que el paciente realice ningún movimiento voluntario de la zona que hay que tratar.
- **Ejercicios Activos.-** son ejercicios en los que el paciente con su propia fuerza de forma voluntaria o autorefleja y controlada, corregidos o ayudados por el fisioterapeuta realiza la rutina de ejercicios.

Dentro de los *Ejercicios Terapéuticos Activos* existen dos subgrupos que dependen del nivel de participación y de independencia que tiene el paciente al momento de ejecutar el ejercicio, estos ejercicios pueden ser *Activos Asistidos* o *Activos Libres*.

- **Ejercicios Activos Asistidos.-** se realizan cuando el paciente no es capaz por sí mismo de llevar a cabo el movimiento en contra de la gravedad, por lo que necesita ayuda para su realización.



- **Ejercicios Activos Libres.**- el paciente ejecuta los movimientos de los músculos afectados exclusivamente sin requerir ayuda, ni resistencia externa.

Considerando que la mayor parte de articulaciones del cuerpo humano son *Articulaciones Sinoviales* (Articulaciones que permiten realizar una amplia gama de movimientos) los ejercicios se componen de movimientos angulares que involucren **abducción** y **aducción**, que implican el movimiento de un hueso alejándose de la línea media y hacia la línea media, ambos se deben realizar en el plano frontal. Además, intervienen movimientos de **flexión** y **extensión**, que consisten en la disminución y aumento del ángulo entre dos huesos.

La extremidad del cuerpo que tendrá movilidad en el ejercicio depende de la articulación que se va a rehabilitar, por ejemplo en las articulaciones de la cadera, hombro y codo la rehabilitación implica los movimientos que se presentan en la Figura 15.

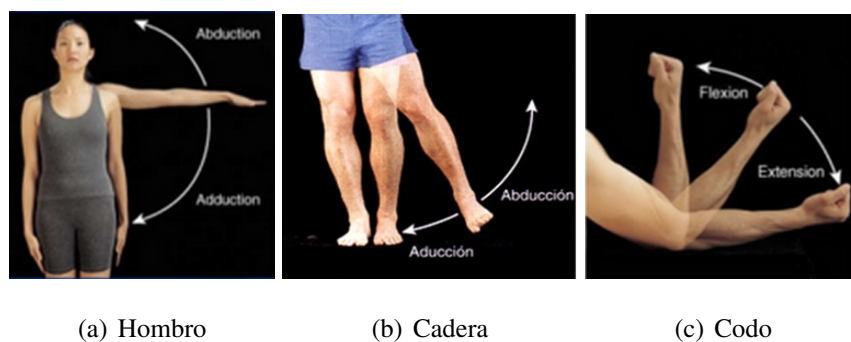


Figura 15: Movimientos para la Rehabilitación de Articulaciones.

**Fuente:** Tortora and Derrickson (2009)

## 2.2 Rehabilitación Virtual

Para obtener la definición de rehabilitación virtual primero se analiza y se parte de los conceptos de rehabilitación y realidad virtual.

- **Rehabilitación.**- Es una terapia orientada a mejorar o restablecer las funciones

del cuerpo humano, de tal manera que el paciente sea reeducado para mejorar las condiciones físicas, sociales o psicológicas que le ha ocasionado la discapacidad producida por algún tipo de traumatismos o alguna enfermedad .

Para recomendar algún tipo de rehabilitación se debe tomar en cuenta las zonas del cuerpo que van a ser tratados para lograr una mejora en la funcionalidad de las mismas, por lo que existen rehabilitaciones o terapias específicas para cada caso. Es así que la rehabilitación se enfoca en:

- Fisioterapia para fortalecer y recuperar la movilidad física.
- Terapia ocupacional para ayudar al paciente en actividades cotidianas y recuperar la condición física.
- Tratamiento de dolor.

Para tratar estos tres aspectos, la rehabilitación se ayuda de las técnicas de fisioterapia y de la planificación de rutinas de ejercicio para el tratamiento de las diferentes patologías de los pacientes.

- **Realidad Virtual (RV).**- las definiciones existentes son numerosas, por lo que se puede decir que, la realidad virtual es una tecnología interactiva que permite recrear ambientes reales con una experiencia sensorial a partir de medios electrónicos. En ALBIOL PÉREZ (2014) se tiene que la RV se compone de las “3i”.
  - Inmersión.- tiene que ver con la capacidad que tienen los usuarios de poder percibir el entorno virtual que lo rodea por medio de los dispositivos y canales sensoriales.
  - Interacción.- consiste en las técnicas para que el usuario tenga una participación activa dentro del entorno virtual.

- Imaginación.- es la capacidad que tiene el desarrollador para solventar problemas a través de las aplicaciones que realiza para los diferentes ámbitos en los que se desenvuelve la sociedad actual.

Además, para tener la certeza de que se esta hablando de Realidad Virtual se debe analizar las consideraciones que se presentan en la Figura 16 que recrea la explicación realizada por Milgram en 1994, la misma que se detalla a profundidad en Milgram and Colquhoun (1999) , en la cual se relaciona la realidad, la virtualización y un nuevo concepto que se obtiene de la combinación de ambas denominado Realidad Mixta en la que se encuentran la Realidad Aumentada y la Virtualización Aumentada.

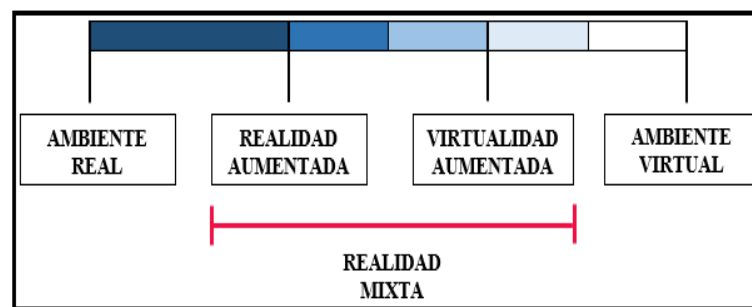


Figura 16: Continuo de la Virtualidad de Milgram.

En los “Ambientes Reales” se tiene la presencia de elementos u objetos únicamente reales sin la implementación de virtualización. Por otro lado, en los “Ambientes Virtuales” se tiene elementos únicamente virtuales. En medio de éstos dos extremos se encuentra la “Realidad Mixta” la cual involucra una fusión de elementos reales y virtuales en tiempo real. Dentro de la “Realidad Mixta” esta la “Realidad Aumentada (RA)” y la “Virtualización Aumentada (VA)”, la primera es una combinación entre elementos reales y virtuales que permiten al usuario interactuar en ambientes que por lo general son tridimensionales; mientras que la segunda, la VA es una inserción de video e imágenes en un entorno compuesto por elementos netamente virtuales.

Por lo tanto, uniendo los dos conceptos de Rehabilitación y Realidad Virtual se define a la ***Rehabilitación Virtual (RHV)*** como la ejecución de sesiones de fisioterapia empleando tecnología de realidad virtual, con el objetivo de satisfacer las necesidades de los usuarios de una manera cómoda y entretenida.

Para realizar las rutinas de *RHV* se ha utilizado dispositivos como Wii, Kinect, Play Station 3 Move e Irex; elementos propios de videoconsolas que con el aporte de la tecnología se han convertido en herramientas poderosas de rehabilitación para personas con discapacidad y están dejando de ser utilizadas únicamente como instrumentos de entretenimiento.

### **2.2.1 Rehabilitación Virtual Motora**

Los trastornos motores que sufren los pacientes están clasificados y son tratados en base a los miembros afectados o que sufren algún tipo de dolencia y en lo relativo al control de la postura y equilibrio.

La rehabilitación motora tradicional consiste en la ejecución de ejercicios repetitivos que muchas de las veces llegan ser cansados, monótono y aburridos. Es por eso que aparece el concepto de *Rehabilitación Virtual Motora (RVM)* que es una tecnología alternativa que consiste en hacer más entretenida la experiencia, ya que tiene un enfoque lúdico involucrando activamente al paciente de tal manera que se genere una motivación, evitando así la falta de interés.

La *RVM* está representando un cambio positivo en las técnicas que comúnmente se utiliza para realizar las sesiones de fisioterapia que buscan mejorar las condiciones de salud de los pacientes y vislumbra un futuro alentador debido a la aceptación que cada vez va en aumento. En ALBIOL PÉREZ (2014) se mencionan las siguientes ventajas que tiene utilizar *RVM*:

- Herramientas de ayuda a los Terapeutas, permitiendo mejorar la atención de los pacientes.

- Parametrización de los sistemas, muchos de los sistemas que se están desarrollando actualmente son además autoadaptativos, siendo capaces de adecuar la dificultad de los ejercicios a lo largo de la sesión en función de cómo lo está haciendo el paciente.
- Seguimiento de los pacientes, almacenándose la actuación y el comportamiento del paciente.
- Uno de los factores críticos dentro de la RVM es la adherencia al proceso terapéutico, y dicha adherencia se ve mejorada significativamente con entornos lúdicos.

### **2.2.2 Rehabilitación Virtual Psicológica**

En el campo de la psicología la aplicación de la RV es muy utilizada para recrear escenarios de la vida real que constituyen las simulaciones empleadas para tratar diferentes tipos de fobia sin poner en peligro la integridad del paciente.

Una de las ventajas que se tiene con la *Rehabilitación Virtual Psicológica* es que las situaciones y elementos que suceden o se presentan en las sesiones están bajo el completo control del terapeuta, lo que genera tranquilidad y confianza por parte del paciente. Además, existen ciertas ventajas que se mencionan en Arbona, García-Palacios, and Baños (2007), las cuales son las siguientes:

- La RV en la psicología permite la graduación de las dificultades en esa práctica, con un nivel de precisión que no permite la realidad.
- Permite repetir exactamente la misma situación las veces que sean necesarias, sin esperar a que esa situación se vuelva a repetir en el mundo real.
- Posibilita realizar una determinada conducta o ejecución en RV sin tener que esperar a que se produzcan determinadas circunstancias en el mundo real.

Por lo tanto, la *Rehabilitación Virtual Psicológica* puede llegar a constituir una herramienta necesaria para ayudar a mejorar las técnicas de exposición que se tiene hasta el momento en el ámbito de la psicología clínica.

### **2.2.3 Impacto Socioeconómico**

Con la implementación de las nuevas técnicas de rehabilitación que propone la *Rehabilitación Virtual* como alternativa a la que se venía utilizando tradicionalmente se puede lograr una mejora en la calidad de vida en los pacientes y/o del impacto económico que puede suponer la implantación de estos sistemas, ya que cuentan con numerosos beneficios como los que se detallan en ALBIOL PÉREZ (2014), entre los que destacan los siguientes:

- Uso de herramientas de bajo coste. Esto permite minimizar los gastos ocasionados a la hora de integrar la tecnología en los Centros Asistenciales.
- Creación de tratamientos más ecológicos, con un mayor impacto en la recuperación. La integración de técnicas de RV permite recrear actividades y situaciones de la vida diaria .
- Permite un seguimiento objetivo sesión a sesión de la evolución del paciente, lo cual no es posible en la rehabilitación tradicional, donde hay que pasar test y escalas explícitamente para saber la evolución del paciente.
- Incremento de los vínculos Terapeuta/Paciente puesto que La integración de herramientas tecnológicas supone un atractivo añadido para los pacientes, respecto a la rehabilitación tradicional.
- Posibilidad de integrar sistemas complementarios on-line (tele-rehabilitación), permitiendo al paciente continuar o complementar el proceso rehabilitador a distancia con el respectivo seguimiento de las actividades foterapéuticas realizadas en el domicilio.

## 2.3 Captura de Movimiento (*MoCap*)

Es una técnicas de grabación de movimientos de personas o animales vivos, para trasladar estos movimientos a modelos digitales. *MoCap* es muy utilizada en actividades cinematográficas, médicas y militares. La tecnología *MoCap* se basa en la generación de imágenes por computador, es por ello que no puede existir sin esta herramienta.

### 2.3.1 Tipos de *MoCap*

Los sistemas *Mocap* disponibles comercialmente se pueden clasificar en tres grupos principales que dependen de la tecnología o sistemas que se utilizan para la captura de movimientos: Sistemas Ópticos, Sistemas Magnéticos y Sistemas Mecánicos. Existen otros sistemas como el Inercial y el ultrasonido, pero son utilizados con muy poca frecuencias en entornos de entretenimiento (Kitagawa & Windsor, 2012).

- **Sistemas Ópticos**

Los sistemas ópticos tienen como componentes principales cámaras *Charge Couple Device* (CCD) que envían datos a un único ordenador, el número de cámaras se comprende habitualmente entre 4 y 32 aunque con dos es suficiente para obtener la información necesaria, ya que el incremento innecesario complica el procesamiento. Son sistemas fiables y permiten capturar movimientos en tiempo real que son utilizados en sistemas de última generación.

Utilizan los datos recogidos por sensores de imagen para lograr identificar la posición de un elemento en el espacio, utilizando una o más cámaras sincronizadas para proporcionar proyecciones simultáneas. Los datos son recogidos a partir de indicadores denominados *markers* que están pegados en el cuerpo de la persona que realiza los movimientos, aunque los sistemas que aparecieron recientemente permiten obtener datos rastreando superficies del sujeto. Dependiendo

del tipo de indicadores que se utilice se tienen las siguientes formas de capturar movimiento con sistemas ópticos.

***Mediante Indicadores Pasivos.***- Los indicadores pasivos están recubiertos de un material reflectante y se pegan al actor en puntos estratégicos, de tal manera que la luz que reflejan se originen cerca de las cámaras y recojan la información. No es necesario colocar dispositivos electrónicos en el cuerpo.

***Mediante Indicadores Activos.***- Los indicadores emiten su propia luz. La posición de los marcadores se puede obtener iluminando uno por cada intervalo de tiempo o varios indicadores a la vez, para lo que se necesita tener todos sincronizados y puedan iluminarse todos en una sola captura.

***Mediante Indicadores Activos Modulados en el Tiempo.***- Se iluminan muchos a la vez mediante luz estroboscópica, se analiza la frecuencia de destello para conocer la ubicación o identidad de cada marcador.

***Mediante Indicadores Semi-Pasivos Imperceptibles.***- Son los propios indicadores los que detectan su propia posición y orientación. Además, permiten reducir considerablemente el tráfico de datos, ya que no utilizan cámaras de alta frecuencia.

***Sin Marcadores.***- no requiere que éstos vistan equipos especiales y utilizan algoritmos que analizan distintas fuentes de entrada de imágenes identificando formas humanas y descomponiéndolas en trozos para realizar el seguimiento de sus movimientos.



- **Sistemas Mecánicos**

En los sistemas mecánicos la captura de movimiento se realiza utilizando sensores mecánicos. Para el proceso de captura de movimientos la persona debe vestir trajes especiales que se adaptan al cuerpo. Los trajes son generalmente estructuras rígidas compuestas de barras metálicas o plásticas unidas mediante potenciómetros colocados en las principales articulaciones, los cuales recogen datos sobre el grado de apertura (Beltrán & López, n.d.).

## **2.4 Kinect**

Es un dispositivo controlador para videojuegos que permite a los usuarios controlar e interactuar con la consola o PC sin la necesidad de ningún periférico adicional mediante la interfaz que capta y reconoce gestos humanos, voz e imágenes. Es por ello que es muy adecuado para desarrollar aplicaciones en educación, deporte, salud, etc. En el ámbito es un dispositivo que podrá ser utilizados en múltiples situaciones debido que no es invasivo.

El sensor *Kinect* fue presnetado por primera vez en junio de 2009 con el nombre de “*Project Natal*”, pero fue en 2010 específicamente el 13 de junio, que se difundió comercialmente con el nombre que se lo conoce actualmente. Desde junio de 2011 esta disponible para PC a través de Windows 7 y Windows 8, permitiendo a los usuarios interactuar con el computador sin necesidad de tener contacto físico.

### **2.4.1 Componentes**

La ubicación de todos los componentes que constituyen el sensor se puede apreciar en la Figura 17.

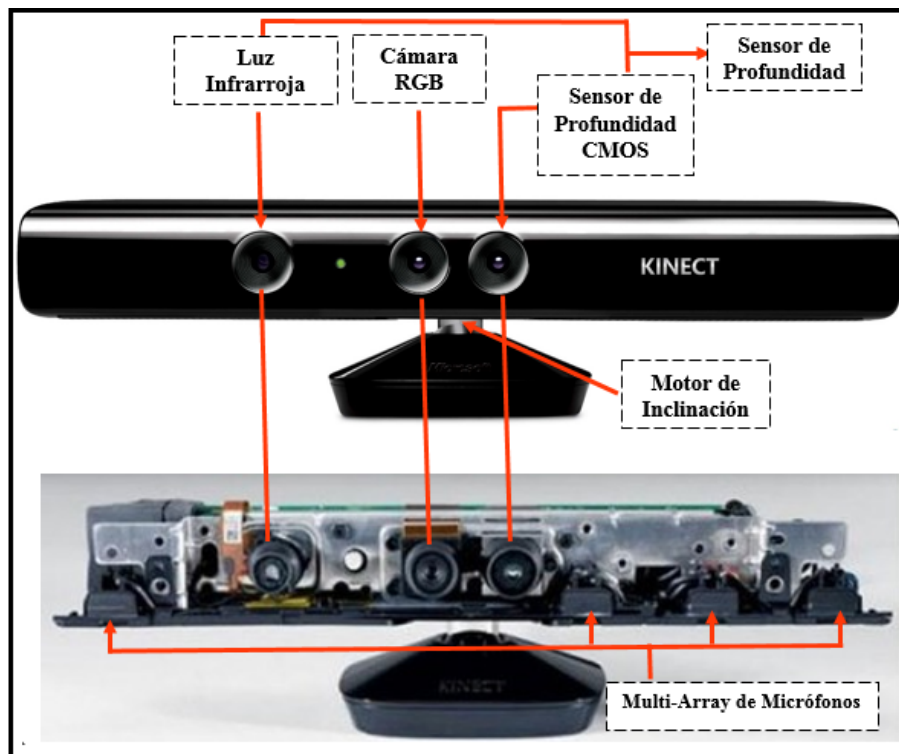


Figura 17: Componentes del Sensor Kinect.

Una breve descripción de cada uno de los componentes se presenta en la Tabla 1.

Tabla 1:

Componentes Internos de *Kinect*.

Componente	Descripción
<b>Ventilador</b>	Permite regular la temperatura en el dispositivo y evitar que algún componente se afecte debido a la disipación de calor.
<b>Acelerómetro</b>	Ayuda a la estabilidad de las imágenes cuando el sensor Kinect se mueve.

CONTINÚA →

Componente	Descripción
<b>Sensor de profundidad</b>	Compuesto de dos partes, un proyector de rayos infrarrojos, que medio de un diodo láser permite obtener un patrón de puntos sobre la superficie que se encuentra frente al sensor y un sensor CMOS monocromático, que genera la información de profundidad a partir de una comparación realizada internamente entre la un patrón capturado con un patrón original. La combinación de los dos elementos que conforman el sensor de profundidad permite realizar el seguimiento del cuerpo humano.
<b>Cámara Color RGB</b>	Permite capturar movimiento y manipular la imagen.
<b>Multi-Array de Micrófonos</b>	Conjunto de cuatro micrófonos que permiten capturar audio. Están colocados estratégicamente para la eliminación de eco y ruido ambiental.
<b>Motor de Inclinación</b>	Permite el movimiento del sensor.
<b>Chip <i>PRIMESENSE PS1080</i></b>	Sirve para reconstruir las imágenes 3D de las escenas que se presentan frente al sensor <i>Kinect</i> . Este chip trabaja en conjunto con el proyector de infrarrojo y el sensor de profundidad.
<b>Memoria RAM</b>	Esta equipado con una memoria RAM de 512 Mb

## 2.4.2 Funcionamiento

El funcionamiento del sensor se basa en la captura de movimientos, siendo capaz de capturar imágenes en 3D mediante la combinación de la información que se obtiene de las cámara RGB y el sensor de profundidad. La información obtenida se denomina RGB-D, ya que posee información de color y de profundidad en una resolución de 320x240, en la cual a cada uno de los pixeles se les asigna información RGB-D.

Para entender el funcionamiento que el sensor *Kinect* realiza al momento de reconocer imágenes se explica a continuación el funcionamiento de la cámara RGB y del Sensor de Profundidad:

- **Cámara RGB.** El proceso de funcionamiento es similar al de las cámaras digitales convencionales, en el que una luz atraviesa un lente que la conduce a un filtro para que pueda ser separada en colores primarios, los mismos que son proyectados a un sensor fotosensible que genera una señal eléctrica en función de la intensidad de la señal. Esta señal, mediante un *ADC*, es convertida en digital, que puede ser almacenada.
- **Sensor de Profundidad.** el sensor emite una constelación de puntos de una luz que es invisible para el ojo humano, cercana en el espectro a los infrarrojos, de esa manera se encuentra el tiempo que tarda la luz en volver al sensor tras reflejarse en los objetos. Finalmente con ésta información y sabiendo la velocidad de la luz se puede tener la distancia a la cual se encuentran los objetos. El sensor *Kinect* es capaz de distinguir la profundidad de los objetos con diferencias de un centímetro, mientras que la altura y ancho con diferencias de tres milímetros.

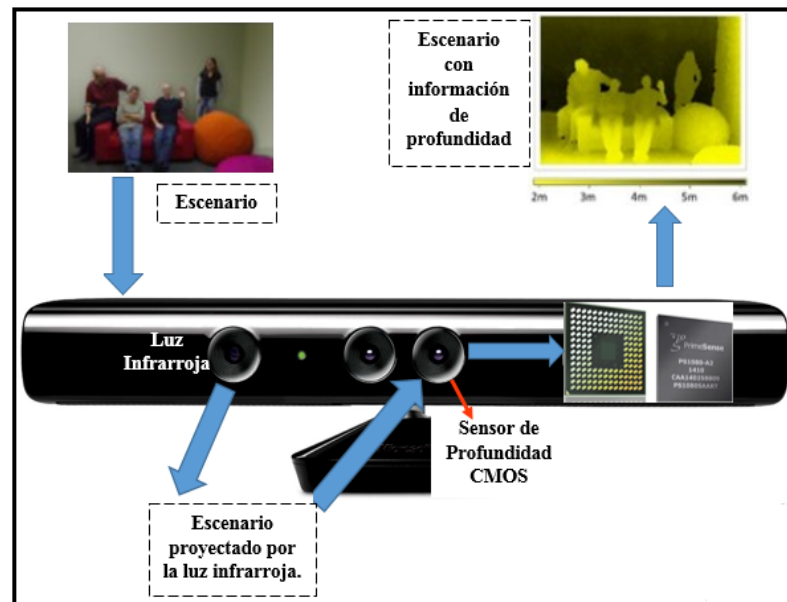


Figura 18: Funcionamiento del Sensor de Profundidad.

Con los datos obtenidos *Kinect* es capaz de ejecutar filtros para identificar si la imagen que se encuentra en frente se trata de una persona, utilizando directrices que se basan en el reconocimiento de la cabeza, dos piernas y dos brazos, que permite diferenciar la imagen en estudio del resto que se encuentran a su alrededor como sillas, mesa, sofás, etc. La información resultante se ordena y se traduce en la identificación de un esqueleto. *Kinect* tiene precargadas más de 200 posiciones comunes del ser humano por lo que en caso de que alguna acción tape alguna parte del esqueleto a la cámara, *Kinect* llenará el vacío automáticamente (Ilvay Today, 2014).

El proceso es continuamente realizado por el sistema a una velocidad de 30 fps (frames por segundo), considerando una ubicación de dos metros para que pueda ser reconocida la imagen.

### 2.4.3 Campo de Visión del Sensor *Kinect*

El campo de visión del sensor *Kinect* depende los datos de profundidad, color y esqueletos.

1. **Visión de Profundidad.**- se considera los puntos de *Limitaciones Físicas*(0.4m a 3m) que corresponde a las capacidades reales del sensor en lo que respecta al área de captura y los *Puntos Dulces* (0.8m a 2.5m) que son las zonas de captura óptimas sin que se encuentren ciertas extremidades del cuerpo fuera del área de captura. En la Figura 19 se puede apreciar de manera gráfica los rangos de visión de profundidad.

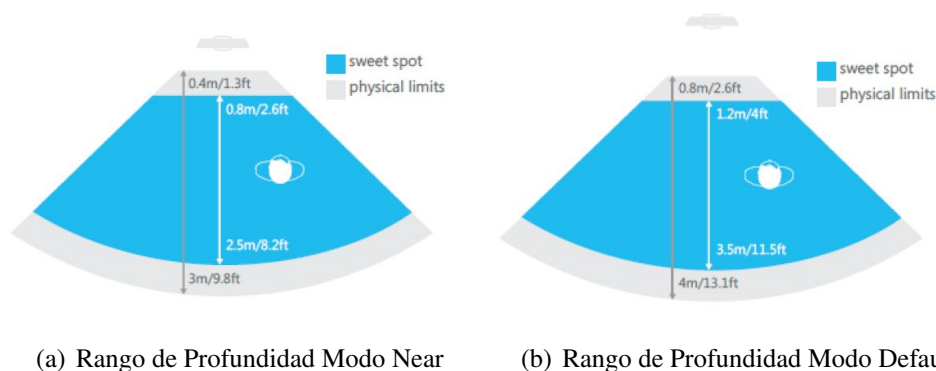


Figura 19: Campo de Visión de Profundidad.

**Fuente:** Rodriguez Estevez (2013)

2. **Visión Vertical y Horizontal.**-En el modo de rango por defecto, Kinect puede ver a la gente de pie entre 0,8 metros (2,6 pies) y 4,0 metros (13,1 pies) de distancia; los usuarios tendrán que ser capaces de utilizar sus brazos a esa distancia, lo que sugiere un rango práctico de 1.2 a 3.5 metros (ver Figura 20).

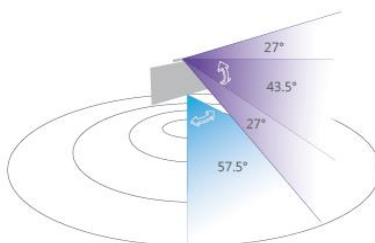


Figura 20: Campo de Visión Horizontal y Vertical.

**Fuente:** Rodriguez Estevez (2013)

#### 2.4.4 Software Controlador del Sensor *Kinect*

Se tiene disponible gran variedad de librerías, cada una con sus *drivers* propios que permiten controlar el sensor. A continuación se presentan las alternativas de desarrollo más utilizadas en la creación de aplicaciones para *Kinect*.

a) ***OpenKinect (Opensource)***

Apareció en 2010 a través de ingeniería inversa, siendo el resultado de los grandes esfuerzos realizados por desarrolladores de todo el mundo para poder interpretar y manipular los datos entregados por el sensor *Kinect*. La comunidad *OpenKinect* tiene como labor principal el desarrollo de *libfreenect*, el paquete software para controlar el sensor *Kinect*, el mismo que permite obtener los datos de imagen a color y el mapa de profundidad.

b) ***OpenNI SDK (Opensource)***

Fue creado por *PrimeSense*, organización sin ánimo de lucro, líder en interfaces naturales y encargada de la fabricación del sensor de profundidad para el dispositivo *Kinect*. Además de proporcionar los datos de bajo nivel (imagen a color y el mapa de profundidad) permite obtener datos de alto nivel, como el seguimiento de cuerpos.

c) **SDK de *Microsoft* para *Windows***

El kit de desarrollo de software con sus siglas en inglés SDK (*Software Developed Kit*) apareció en 2012, dos años después de la aparición del *Kinect* y fue desarrollado por *Microsoft* debido al interés que generó el sensor. El SDK oficial incluye *drivers* para usar la *kinect* en *Windows*, una API, documentación, códigos de ejemplo y diversas herramientas bastante útiles para el desarrollo de aplicaciones que impliquen el *Kinect* (Rodríguez Estevez, 2013).

Ofrece a los usuarios, programadores y estudiantes gran cantidad de opciones de desarrollo para conseguir aplicaciones innovadoras y así aprovechar todo el potencial que tiene el sensor, ya que se puede implementar en diferentes áreas.

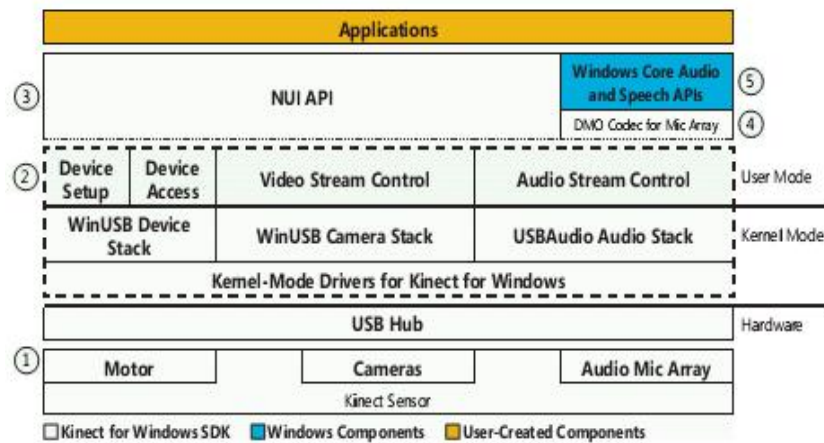
## ARQUITECTURA

La arquitectura del SDK de *Kinect* oficial de Microsoft consta de tres capas, las cuales se comunican entre sí dependiendo de los requerimientos de la aplicación que se va a desarrollar. A continuación se explicará de abajo hacia arriba la estructura de capas de la arquitectura.

- En el nivel más bajo se encuentra el sensor *Kinect*, ya que el hardware comprende todos los sensores y cámaras.
- En la capa siguiente se encuentran todos los controladores que permiten la programación, ya que es en esta capa en la que se digitaliza los datos provenientes de los sensores.
- La capa inmediatamente superior a la anterior se encarga de establecer la conexión entre la información del hardware del sensor *Kinect* y las APIs.
- La última capa corresponde a las APIs, una para cada tipo de desarrollo, como por ejemplo lenguajes de programación como C#, Visual Basic, JavaScript y C++ con XAML.

La arquitectura completa del SDK de *Microsoft* es la que se presenta en la Figura 21 . Los detalles de los campos se explican a continuación de la figura en la Tabla 2.



Figura 21: Arquitectura del SDK de *Microsoft*

Fuente: Rodríguez Estevez (2013) Microsoft (n.d.)

Tabla 2:

Explicación de los Campos de la Arquitectura del SDK de *Kinect*.

Nº	Campo	Descripción
1	<b>Hardware</b>	Corresponde a todos los componentes de <i>Hardware</i> , incluyendo al sensor <i>Kinect</i> y al conector USB mediante el cual el dispositivo está conectado a la PC.
2	<b>Drivers</b>	Corresponde a los controladores de audio y de transmisión de video para el respectivo streaming de datos, así como los controladores de enumeración de dispositivos <i>Kinect</i> .
3	<b>Interfaz de Usuario Natural de Kinect para Windows(NUI)</b>	A través de ella se puede acceder a los datos de audio, vídeo, profundidad y esqueletos a través de la aplicación.

CONTINÚA →

N°	Campo	Descripción
4	<b>DirectX Medios Object (DMO)</b>	Sirve para la localización de las fuentes de audio y para poder utilizar la capacidad del entorno gráfico en la aplicación.
5	<b>Windows APIs estándar</b>	El audio, voz y APIs multimedia de Windows.

Al relacionar los diferentes niveles de la arquitectura y al tener acceso a ellos el SDK proporciona una biblioteca de software sofisticado y herramientas para ayudar a desarrollar aplicaciones utilizando una amigable forma de ingreso natural basado en Kinect, que detecta y reacciona a los acontecimientos del mundo real.

## 2.4.5 Sistemas de Reconocimientos de Gestos y Posturas

En la actualidad se tiene tres principales sistemas de reconocimiento de gestos y posturas mediante Kinect, a continuación se realiza una explicación de cada uno de ellos.

### 2.4.5.1 Reconocimiento Basado en Coordenadas

Es el sistema más básico, su funcionamiento se basa en analizar los gestos que se van a usar en el sistema y codificarlos directamente realizando comparaciones entre las coordenadas del cuerpo para verificar que están cumpliendo las condiciones necesarias para cada gesto.

En la Figura 22 se puede apreciar un ejemplo de la implementación de éste tipo de sistema de reconocimiento de gestos y posturas mediante *Kinect*. Se puede observar el desplazamiento a la derecha de la mano partiendo desde el hombro, comparando las coordenadas xy en la posición inicial y final del movimiento.

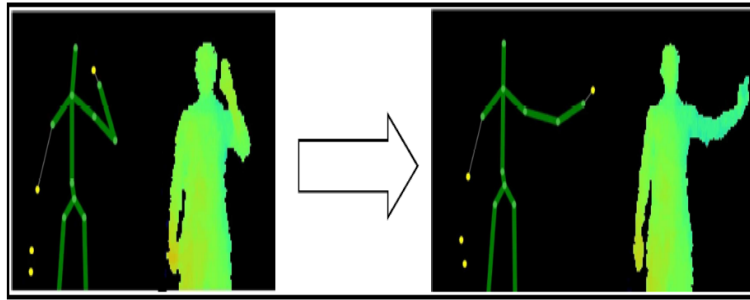


Figura 22: Desplazamiento a la derecha.

**Fuente:** Robledo Zarco (2012)

#### 2.4.5.2 Basado en Modelos Ocultos de Markov

Son sistemas basados en un modelo estadístico que dada la información de entrada trata de extraer los parámetros ocultos que podrían componer el patrón de entrada. Esto los hace muy eficientes para realizar después análisis de patrones lo que los hace idóneos para el reconocimiento de gestos y posturas, ya que estos no son más que sucesiones de coordenadas siguiendo unos determinados patrones, diferentes para cada gesto (Robledo Zarco, 2012).

#### 2.4.5.3 Reconocimiento Basado en un algoritmo de “*Dynamic Time Warping*”

Esta técnica permite reconocer patrones lineales en el tiempo que se produzcan a velocidades diferentes a la velocidad en la que fueron capturados. Estos sistemas tienen la principal ventaja de que permiten un porcentaje de acierto muy alto aun variando la velocidad de realización del gesto. El principal problema que tiene este modelo es que aunque abstrae muy bien la velocidad de ejecución del gesto, tiene más problemas con las desviaciones sobre el mismo, y dado que lo interesante es conseguir un reconocimiento de patrones lo más universal posible, esta alternativa pierde algo de interés.

## 2.5 Base de Datos

Las bases de datos son sistemas computarizados que cumplen con funciones de almacenamiento y organización de registros, para que de esa manera los usuarios puedan modificar y recuperar la información almacenada.

Para realizar las diferentes acciones el usuario tiene las siguientes operaciones básicas que puede realizar en la base de datos:

- Insertar Datos.
- Modificar Datos.
- Eliminar Datos.

En la Figura 23 se presentan los conceptos básicos de las bases de datos que se debe conocer para el desarrollo del presente proyecto.

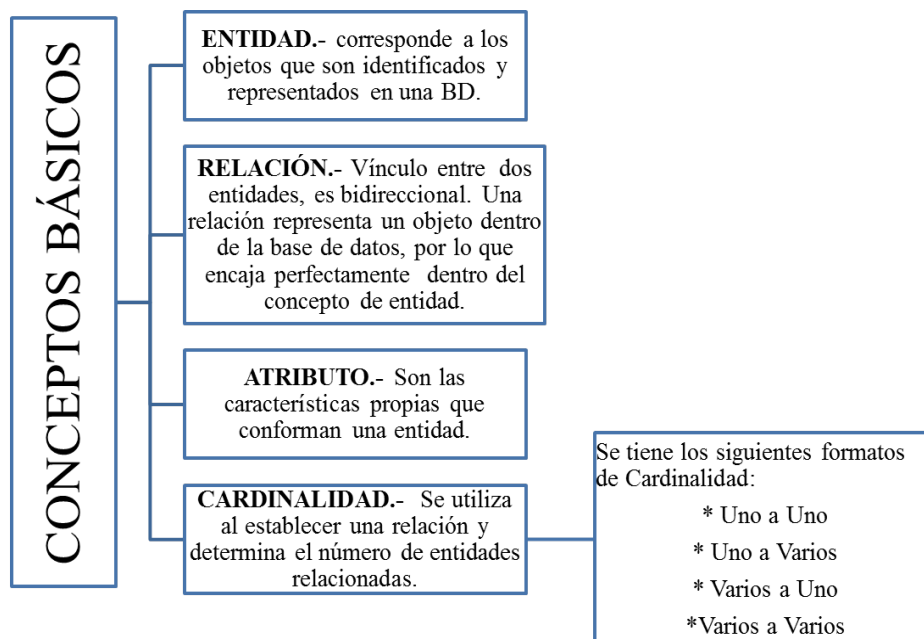


Figura 23: Conceptos Base de Datos.

El componente más importante en una base de datos es el **Sistema de administración de base de datos (DBMS)**, que viene a ser la interfaz usuario-BD y con el

cual se puede llevar a cabo las operaciones de inserción, eliminación y modificación de datos.

### 2.5.1 Base de Datos MySQL

Las bases de datos se utilizan en todas las áreas negocios, salud, educación, etc. Muchas empresas utilizan las bases de datos para realizar un seguimiento de la información sobre sus clientes, lo que ha contribuido para el crecimiento explosivo de Internet. La mayor parte de aplicaciones web han desarrollado sus bases de datos empleando MySQL (Suehring, 2002).

MySQL es un sistema de administración de bases de datos de código abierto, ofrece disponibilidad de código fuente, portabilidad, fácil aprendizaje y lo más importante alto desempeño. Por las características que tiene se ha convertido en el gestor de base de datos más popular en el mundo, no solo en empresas o desarrollos pequeños sino que grandes corporaciones como Google, Yahoo, CISCO, Motorola, Texas Instruments, entre otras, han aplicado MySQL para implementar sus bases de datos (Cobo, 2005).

Al estar basado en SQL, que es un lenguaje declarativo más utilizado para acceder a datos relacionales que expone lo que requiere y no indica cómo hacerlo, las acciones de inserción, modificación, selección y eliminación se realiza con sentencias SQL (Cobo, 2005). Las sentencias son las siguientes:

- **INSERT.-** permite ingresar un registro.
- **SELECT.-** permite consultar un registro.
- **UPDATE.-** permite modificar un registro.
- **DELETE.-** permite eliminar un registro.

## 2.6 Lenguaje PHP

PHP es un lenguaje eficiente de tipo script orientado a servidores y fue diseñado para la creación de sitios web, posee varias herramientas útiles incorporadas. Ejemplos de estas herramientas son: generar imágenes GIF, conectar con otros servicios de red, generar documentos e incluso enviar emails.

El lenguaje PHP se caracteriza por su robustez, potencia, versatilidad, rapidez, aprendizaje sencillo y sobre todo es multiplataforma. hay que destacar que los programas escritos bajo este lenguaje pueden también estar embebidos con HTML, ya que tiene un intérprete para poder mostrar al usuario la solicitud realizada en formato HTML (Cobo, 2005).

Por ser un lenguaje del lado del servido se requiere un servidor web y PHP puede trabajar con la totalidad de servidores más conocidos como Apache y los de Microsoft, aunque es muy común encontrar sobre servidores Apache. Los sistemas operativos que permiten PHP también son muy variados, entre los que se encuentran Linux, Unix, Windows, Mac, entre otros.

Por todas las ventajas que ofrece es uno de los lenguajes más utilizados en el mundo para el desarrollo de páginas dinámicas personales y de grandes empresas, ya que también permite la conexión y gestión de bases de datos.

## CAPÍTULO 3

# DISEÑO E IMPLEMENTACIÓN

Este capítulo es el núcleo del trabajo, en el cual se encuentra detallado todo el proceso realizado, desde el diseño inicial hasta la etapa de implementación del prototipo.

### **3.1 Diseño del Prototipo Auxiliar para Fisioterapia.**

En este punto se puede obtener una perspectiva general del sistema, ya que es aquí en donde se define el funcionamiento básico y las restricciones que se presentan a partir de la elaboración del diagrama de bloques, de flujo y de casos de uso. El proceso se lleva a cabo considerando la facilidad que se ofrece para el manejo del sistema por parte del terapeuta y el paciente para una mejor ejecución del proceso de rehabilitación.

#### **3.1.1 Descripción del Sistema.**

El prototipo ha diseñarse se basará en una aplicación que permitirá la interacción entre el usuario y la interfaz mediante el sensor *Kinect*, el mismo que permite obtener la información del movimiento que será procesada con ayuda de programación orientada a objetos C#. La información relevante de la sesión será almacenada en una base de datos para ser recuperada desde una aplicación web en la que el terapeuta pueda visualizar el progreso y cumplimiento de la rehabilitación.

### 3.1.2 Requisitos del Prototipo.

El prototipo auxiliar de fisioterapia debe contar con ciertos elementos que intervienen en su estructura para lograr obtener el funcionamiento que se propone en el apartado anterior correspondiente a la descripción del sistema. Por lo se debe considerar los siguientes requisitos.

- Sensor Kinect necesario para la captura de movimientos.
- Computador con un procesador de 2.66 GHz o superior para el procesamiento de la información.
- Kit de Desarrollo para Kinect que provea de las librerías y que cuente con la documentación necesaria para el desarrollo del proyecto.
- Base de Datos
- Servidor

### 3.1.3 Diagrama de Bloques del Prototipo.

Para el desarrollo del presente prototipo se ha utilizado el diagrama de bloques que se muestra en la Figura 24.

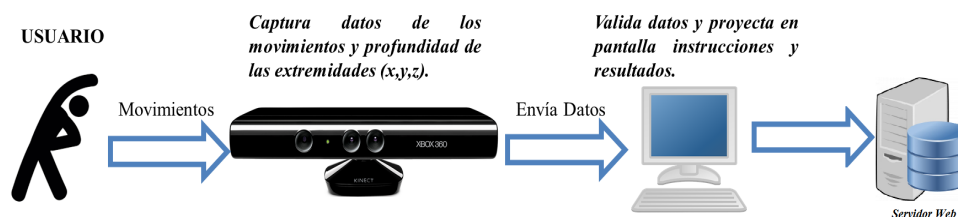


Figura 24: Descripción del Sistema Auxiliar de Fisioterapia.

- **Captura de Datos.-** permite obtener la información de los movimientos realizados por el paciente.



- **Procesamiento de la Información.-** realiza los cálculos y análisis en base a la información proporcionada por el sensor para estructurar la interfaz de usuario y las validaciones necesarias del prototipo.
- **Servidor Web.-** permite implementar la base de datos y la aplicación web.

### 3.1.4 Diseño de la Etapa de Captura de Datos.

En esta etapa es en la que se obtienen la información de los movimientos mediante técnicas de MoCap, utilizando el *Kinect*. El elemento indispensable para la captura de datos es el sensor, por lo que es importante definir el modelo de sensor y el Kit de Desarrollo de Software que se utilizará para el desarrollo del proyecto, para posteriormente establecer el diagrama de bloques que tendrá este módulo.

#### 3.1.4.1 Requisitos de la Etapa de Captura de Datos.

De acuerdo al diseño que se propone, en esta etapa se debe disponer de los siguientes requisitos:

- Sensor Kinect.
  - Cámara RGB.
  - Sensor de Profundidad (cámara y proyector de infrarrojos).
  - Arreglo de Micrófonos.
  - Motor para el ajuste del ángulo de elevación del sensor  $\pm 27^\circ$ .
  - Ángulo de Visión de  $57^\circ$  horizontal y  $43^\circ$  vertical.
  - Rango de distancia de 0.8 m a 4 m.
  - Tensión de Alimentación 12 V.
- Librerías propias de Kinect que permitan:
  - Imagen RGB.

- Trackeo de las articulaciones.
- Reconocimiento de Voz.

### 3.1.4.2 Selección del Sensor Kinect.

Al momento del desarrollo de este trabajo se cuenta con tres alternativas de sensor: Kinect V1 (XBOX 360), Kinect V2 (XBOX ONE) y Kinect para Windows, los tres son completamente útiles para la implementación del prototipo, ya que cuentan con los adaptadores para la PC y las librerías necesarias para el funcionamiento. Se han evaluado las características que presenta cada uno en base a los requisitos planteados para esta etapa y así poder realizar la selección del sensor, en la Tabla 3 se presentan los aspectos que cumplen cada uno de los modelos de sensor.

Tabla 3:

Características de las Versiones de Kinect.

Característica	XBOX 360	XBOX ONE	WINDOWS
<b>Cámara RGB</b>	SI	SI	SI
<b>Sensor de Profundidad</b>	SI	NO	SI
<b>Arreglo de Micrófonos</b>	SI	SI	SI
<b>Ajuste de Ángulo de Elevación (<math>\pm 27^\circ</math>)</b>	SI	NO	SI
<b>Ángulo de Visión (<math>57^\circ</math> horizontal y <math>43^\circ</math> vertical)</b>	SI	NO	SI
<b>Rango de Distancia (0.8 m a 4 m)</b>	SI	NO	SI
<b>Detección de Dedos y Muñecas</b>	NO	SI	NO
<b>Detección de músculos</b>	No	Si	No
<b>Costo</b>	\$ 150	\$ 150	\$ 250

De acuerdo a la Tabla 3 de selección, los tres modelos cumplen con el requisito de la cámara de RGB, el sensor de profundidad y el arreglo de micrófonos. En el casos

del sensor Kinect para XBOX One presenta teóricamente mejores prestaciones para el desarrollo del proyecto, pero el problema radica es que no se han tenido experiencias anteriores que hayan utilizado este sensor para el desarrollo de las aplicaciones y no se conoce la estabilidad del mismo. Por lo que queda únicamente las otras dos opciones, que si bien es cierto tienen características similares, el costo es un factor determinante al momento de la selección. Además, se debe considerar que el *Kinect* para Windows no se tiene fácilmente disponible en el mercado ecuatoriano, por lo que se debe adquirir directamente de la página oficial de la tienda de Microsoft e implicaría valores agregados al costo del sensor y tiempo. Por todas las razones mencionadas se ha seleccionado el sensor *Kinect* para XBOX 360, considerando también que proyectos previos mencionados en el estudio del estado del arte han utilizado este sensor para desarrollar sus aplicaciones y han evidenciado un funcionamiento óptimo del sensor.

#### **Sensor Kinect XBOX 360.**

El sensor para XBOX360 que se presenta en la Figura 25 cuenta con un motor para regular el ángulo de elevación del sensor, un arreglo de micrófonos con el que se puede realizar el reconocimiento de voz, una cámara RGB y un sensor de profundidad que resulta de la combinación de una cámara y un proyector de infrarrojo. Para poder ser conectado a la PC dispone de un adaptador para la regulación del voltaje. Tiene un rango teórico de distancia de 0.8 m a 4 m aproximadamente, puede reconocer teóricamente hasta cuatro personas y veinte puntos del cuerpo humano. Con este sensor el trackeo específico de las articulaciones de la mano no es posible.



Figura 25: Sensor Kinect XBOX360.

### 3.1.4.3 Selección del Kit de Desarrollo de Software.

Una vez seleccionado el sensor *Kinect* XBOX 360 y considerando que existe un gran desarrollo sobre Windows, se ha puesto como criterio de selección del kit desarrollo el implementar el prototipo sobre este mismo sistema operativo. Para lo cual, se realizó un análisis de las características que ofrecen las alternativas open source y la versión oficial de Microsoft, las mismas que se presentan en la Tabla 4.

Tabla 4:

Comparación entre las Alternativas OpenSource y el SDK de Microsoft.

Característica	SDK Microsoft	OpenSource
Facilidad de Adquisición	SI	NO
Facilidad de Instalación	SI	NO
Acceso a la Documentación Completa y a Ejemplos.	SI	NO
Sistema Operativo Windows	SI	SI
Precisión y Velocidad en el Trackeo de Articulaciones	SI	NO
Soporte a Múltiples Sensores	SI	NO

Por los aspectos evaluados y presentados en la Tabla 4 se considera que para el desarrollo del proyecto la mejor opción es el **SDK de Microsoft**, ya que cuenta con soporte, herramientas y gran cantidad de documentación que puede servir de ayuda para el cumplimiento de los objetivos. Pero sobre todo la selección se realizó considerando que el sensor y el kit de desarrollo son productos de la misma empresa, por lo que ofrece mayor estabilidad y las actualizaciones necesarias. Además de las características mencionadas, el SDK también ofrece datos de profundidad, cámara de color, audio y esqueletos, los cuales se pueden manipular dependiendo de la aplicación.

El uso de las alternativas Open Source no aportan aspectos importantes para el

desarrollo del proyecto, más bien dificultarían el proceso al no contar con la documentación completa para lograr una mejor comprensión. La única característica importante que tiene es la de ser multiplataforma, que en el caso del prototipo de fisioterapia no es un requisito necesario, ya que se va a trabajar sobre Windows.

#### 3.1.4.4 Diagrama de Bloques de la Etapa de Captura de Datos.

Una vez establecidos el sensor y las librerías que se van a utilizar, se propone el diagrama de bloques de la Figura 26 que describe el proceso de captura de datos por parte del sensor *Kinect*.

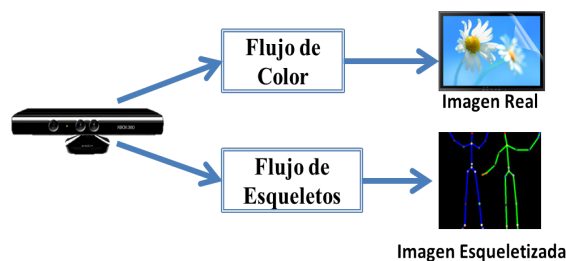


Figura 26: Diagrama de Bloques de la Etapa de Captura de Datos.

El sensor captura los datos mediante las cámaras de color y profundidad que posee, las mismas que generan flujos de datos que permiten obtener la imagen real de la escena y la esqueletización del cuerpo humano, a estos flujos de datos se puede acceder con la ayuda de las funciones propias del SDK de Microsoft.

- **Flujo de Color.-** este flujo permite obtener la imagen real de la escena, al habilitar se puede acceder al formato que tiene la captura de datos mediante *ColorImageFormat* que permite siete formatos diferentes, siendo el RGB Resolution 640x480 Fpd30 el que se establece automáticamente sin que se haya definido uno en particular. Cada pixel de los frames que se generan contienen cuatro bytes (ver Figura 27), un byte por cada elemento de color que se controlan en el formato RGB(Rojo, Verde, Azul y un nivel de transparencia). De esa manera se puede manipular los valores de cada byte del pixel modificando los colores de la imagen.

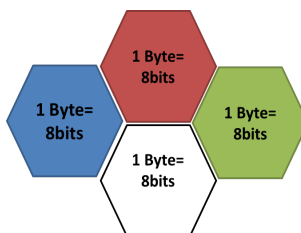


Figura 27: Estructura del Un Pixel en Formato RGB.

- Flujo de Esqueletos.-** Un esqueleto humano se compone de huesos y articulaciones (Bones and Joints, en inglés) y tiene alrededor de 206 huesos, que es una gran cantidad de información para programar, para eso el SDK de *Kinect* ha simplificado el trabajo a 19 huesos y 20 articulaciones, obteniendo cada una de ellas mediante una en una combinación de las cámaras RGB y de profundidad. El manejo de estas articulaciones o Joints se debe realizar con sus nombres en inglés, ya que desgraciadamente los desarrolladores de los lenguajes de programación y las nuevas tecnologías son empresas y personas que manejan este idioma. En la Figura 28 se tiene una lista con los nombres en inglés de las 20 articulaciones que están disponibles en el SDK de Kinect.

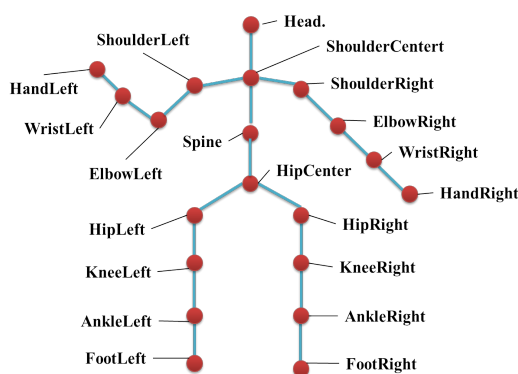


Figura 28: Lista de Articulaciones del SDK de *Kinect*.

Mediante funciones que proporciona el SDK es posible obtener las coordenadas 3D de las articulaciones. En la Figura 29 se puede apreciar el plano tridimensional que utiliza el sensor para conocer las coordenadas de las articulaciones.

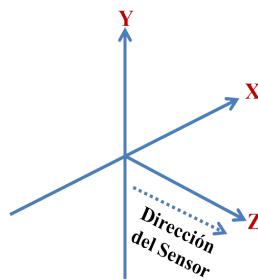


Figura 29: Plano de Referencia de las Coordenas de las Articulaciones.

### 3.1.5 Diseño de la Etapa de Procesamiento de la Información.

Es la etapa más extensa en el desarrollo del proyecto, ya que contempla la parte más importante y fundamental del prototipo, la interfaz gráfica de usuario (GUI). Es aquí en donde se establecen las validaciones y se definen los movimientos que permiten realizar las sesiones de rehabilitación. Además, se establece también la conexión a la base de datos y por ende a la aplicación web.

#### 3.1.5.1 Requisitos de la Etapa de Procesamiento de la Información

De acuerdo al diseño que se propone, los requisitos que se contemplan en esta etapa guardan relación con la interfaz gráfica (software) del prototipo y son los siguientes:

- Debe disponer de Autenticación para llevar a cabo el control de los usuarios.
- Permitir el registro de usuarios en el sistema.
- Interactuar con el sensor *Kinect*.
- Procesamiento y validación de movimientos.
- Conexión a la base de datos.
- Interfaz gráfica intuitiva y amigable para el usuario.

### 3.1.5.2 Diagrama de Bloques de la Etapa de Procesamiento de la Información

El diagrama de bloque que describe la estructura de esta etapa se presenta en la Figura 30.

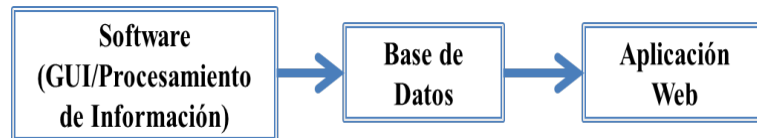


Figura 30: Diagrama de Bloques de la Etapa de Procesamiento.

- **Software.-** establece la estructura que tiene la interfaz de usuario del prototipo, en base a la información proporcionada por el sensor. Tiene conexión con la base de datos para almacenar los datos de la sesión de rehabilitación y que esta a su vez establece comunicación con la aplicación web para presentar los parámetros enviados desde la aplicación correspondientes a la rutina de ejercicios realizada por el paciente.
- **Base de Datos.-** almacena la información de la sesión de rehabilitación.
- **Aplicación Web.-** permite visualizar la información de la sesión de rehabilitación.

En este apartado se realiza el diseño del Software que comprende la interfaz gráfica del prototipo, ya que el diseño de la base de datos y de la aplicación web forman parte de la etapa del Servidor Web.

### 3.1.5.3 Requisitos de la Interfaz Gráfica de Usuario (GUI).

El prototipo auxiliar de fisioterapia debe identificar los movimientos a través del sensor *Kinect*, procesar la información de tal manera que se pueda determinar si el movimiento realizado por el paciente es válido y cuenta como una repetición, a partir de la comparación con un patrón de movimientos generados por el fisioterapeuta. Los ejercicios con los que cuenta el prototipo permiten rehabilitar las articulaciones



del hombro, cadera y codo, los movimientos consisten en abducción, aducción, flexión y extensión, los cuales el paciente debe realizar sin ayuda externa por lo que los ejercicios terapéuticos son de tipo activos libres. Además, el prototipo permite enviar parámetros como el tiempo de ejecución de la rutina y el número de repeticiones de los movimientos a un servidor para que sean almacenados en una base de datos y así poder llevar un historial y determinar el progreso de la rehabilitación.

Con el fin de lograr un correcto funcionamiento de la aplicación varios requerimientos deben ser analizados, por que han sido divididos en dos grupos *Funcionales* y *No funcionales*. Dentro del primer grupo se encuentran los requisitos que describen el funcionamiento del sistema, mientras que en segundo grupo están los requisitos especifican criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos y pueden ser de algunos tipos como: rendimiento, interfaz, operación, recursos, comprobación, documentación, seguridad, calidad y mantenimiento.

Además, para realizar una adecuada especificación de requisitos de software (ERS) se debe considerar algunas características definidas en el estándar IEEE 830-1998, en el que destaca que una buena ERS debe ser completa, consistente, inequívoca, correcta y priorizable.

En las tablas(Tablas 18- Tabla 9) que se presentan a continuación se detallan los requisitos del sistema tanto funcionales como no funcionales. Para poder identificar a cada requisito se ha utilizado la nomenclatura que se tiene en la Tabla 5.

Tabla 5:

Nomenclatura de los Tipos de Requisitos de la Aplicación.

<b>Identificador</b>	<b>Tipo de Requisito</b>
<b>RF</b>	Requisito Funcional.
<b>RNFR</b>	Requisito No Funcional de Rendimiento.
<b>RNFI</b>	Requisito No Funcional de Interfaz.
<b>RNFO</b>	Requisito No Funcional de Operación.
<b>RNFC</b>	Requisito No Funcional de Comprobación.
<b>RNFD</b>	Requisito No Funcional de Documentación.
<b>RNFM</b>	Requisito No Funcional de Mantenimiento.
<b>RNFU</b>	Requisito No Funcional de Usabilidad.
<b>RNFS</b>	Requisito No Funcional de Soporte.

**Fuente:** Fernández Sánchez (2012)

Además, en cada tabla se encuentra un Identificador considerando la nomenclatura antes mencionada, Título que es el nombre del requisito y una breve descripción del mismo. También se ha considerado que cada requisito dependiendo del tipo tiene una *Prioridad, Estabilidad y Necesidad* alta o esencial, lo que significa que estos requisitos son importantes y que no puede ser modificado en el desarrollo del proyecto.

## REQUISITOS FUNCIONALES

Tabla 6:

Requisitos Funcionales del Sistema.

<b>Identificador</b>	<b>Título</b>	<b>Descripción</b>
RF01	Iniciar Aplicación	El usuario deberá iniciar la aplicación
RF02	Ver Opciones de Aplicación en el Menú Principal	Aparece en pantalla una lista con las opciones iniciales del sistema (Ejecutar, Información y Ayuda).
RF03	Selección de Opción de Operación	El usuario podrá seleccionar la opción correspondiente.
RF04	Cerrar Aplicación	El sistema cierra la aplicación, mostrando un mensaje para asegurarse que desea salir.

Tabla 7:

Requisitos Funcionales de la Opción "Ejecutar".

<b>Identificador</b>	<b>Título</b>	<b>Descripción</b>
RF05	Registrarse	Después de seleccionar el modo de operación el usuario debe registrarse para realizar o agregar ejercicios.
RF06	Ver Menú	Se presenta una lista de ejercicios disponibles.

CONTINÚA 

<b>Identificador</b>	<b>Título</b>	<b>Descripción</b>
RF07	Seleccionar Opción	El usuario ingresa en una ventana en la que se presenta las actividades a realizar dependiendo si seleccionó algún ejercicio de la lista.
RF08	Regresar al Menú	El usuario podrá salir de la opción Ejecutar y puede seleccionar otra del menú principal.

Tabla 8:

Requisitos Funcionales de la Opción "*Realizar Ejercicios*".

<b>Identificador</b>	<b>Título</b>	<b>Descripción</b>
RF09	Realizar Movimientos Indicados en pantalla	Los movimientos son comparados con un patrón almacenado que sirve como guía para la validación.
RF10	Finalizar Rutina	El usuario completa el número de repeticiones asignadas a la rutina y finaliza, mostrando los resultados alcanzados.
RF11	Regresar al Menú de Ejercicios	Regresara a la lista de ejercicios para poder seleccionar otro o realizar alguna otra operación disponible.
RF12	Cerrar Opción Realizar Ejercicios	El usuario puede salir de la opción en cualquier momento.

## REQUISITOS NO FUNCIONALES

Tabla 9:

Requisitos No Funcionales de la Aplicación.

<b>Identificador</b>	<b>Título</b>	<b>Descripción</b>
RNFR01	Usuarios en el Sistema	Por utilizar un kinect el sistema admite un usuario a la vez.
RNFI01	Interfaz Gráfica	La aplicación utilizará colores e imágenes.
RNFI02	Contenido del Menú	Debe tener las opciones e instrucciones claras y precisas para no confundir al usuario.
RNFO01	Acceso a la Aplicación	Podrá ser llevado a cabo a través de un archivo ejecutable.
RNFU01	Facilidad de Manejo	Se puede utilizar el sistema sin contar con conocimientos avanzados en computación
RNFS01	Plataforma	El sistema funciona en el Sistema Operativo Windows en versiones posteriores a Windows XP y que tengan instalado el SDK de Kinect.

### 3.1.5.4 Diagrama de Flujo de la Interfaz Gráfica de Usuario (GUI).

En base a los requisitos que debe tener la aplicación se ha realizado el diagrama de flujo de la Figura 31 que describe el funcionamiento de la aplicación.

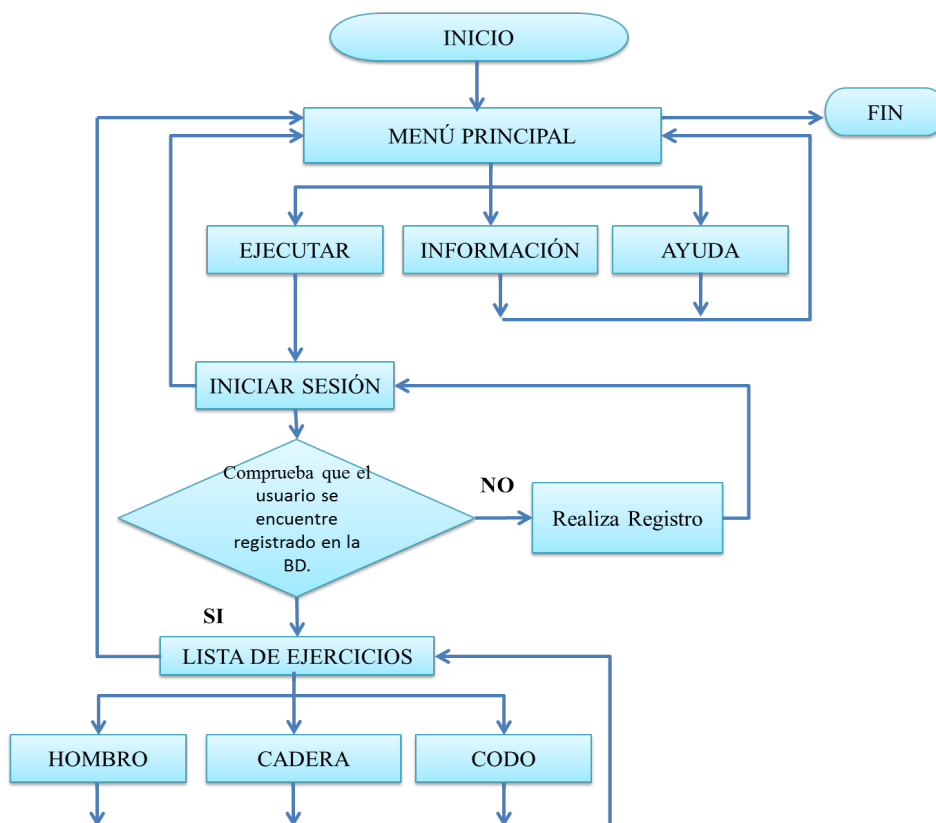


Figura 31: Diagrama de Flujo de la Aplicación.

La interpretación de los bloques resulta muy intuitiva, por lo que únicamente se procede a realizar la explicación del bloque que se encuentra inmerso dentro de las opciones de rehabilitación: Hombro, Cadera y Codo presentes en el diagrama de la Figura 31 y que consiste en **Realizar Ejercicio**, tomando en cuenta que el procedimiento es el mismo para las tres alternativas.

### DESCRIPCIÓN MÓDULO REALIZAR EJERCICIO.

Para realizar el ejercicio el usuario debe realizar los movimientos que se presenta en una imagen de muestra, de los cuales mediante el sensor *Kinect* se obtiene la información que sirve para calcular los ángulos de las articulaciones y así poder establecer, en base a comparación de ángulos, si ha sido ejecutado adecuadamente para poder incrementar el número de repeticiones. Si el usuario no a logrado realizar el movimiento

de manera adecuada, debe realizar nuevamente hasta que pueda ser una postura válida. La rutina finaliza cuando el usuario no desea completar un número mayor de repeticiones y decide salir de la interfaz.

El diagrama de flujo que describe el procedimiento contemplado para realizar el ejercicio se presenta en la Figura 32.

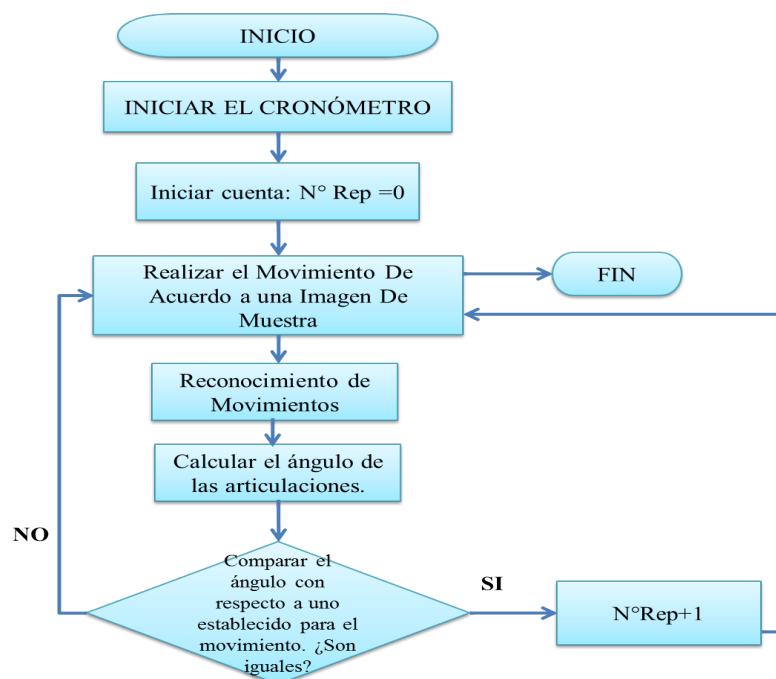


Figura 32: Diagrama de Casos de Uso.

### 3.1.5.5 Diagrama de Casos de Uso de la Interfaz Gráfica de Usuario (GUI).

En el modelo de casos de uso se establece un diagrama que incluye las funciones de la aplicación y los actores que participan en cada uno de los procesos. Para la elaboración del diagrama de casos de uso hay que definir los actores que intervienen con el papel que desempeñan y las necesidades que cada uno tenga del sistema. Hay que tener en cuenta que los casos de uso especifican el comportamiento que debe tener cada proceso y no explica como llevarse a cabo cada uno.

En la Figura 33 se tiene el modelo de casos de uso del sistema, en el que se puede

apreciar que el actor principal del sistema es el **paciente** que realiza la ejecución de los ejercicios de rehabilitación, el mismo que puede acceder a la aplicación personalmente en su hogar o con la ayuda de un fisioterapeuta que será el encargado de supervisar el proceso. Además, se tiene el intercambio de información que tiene el paciente con la aplicación y ésta a su vez con el servidor.

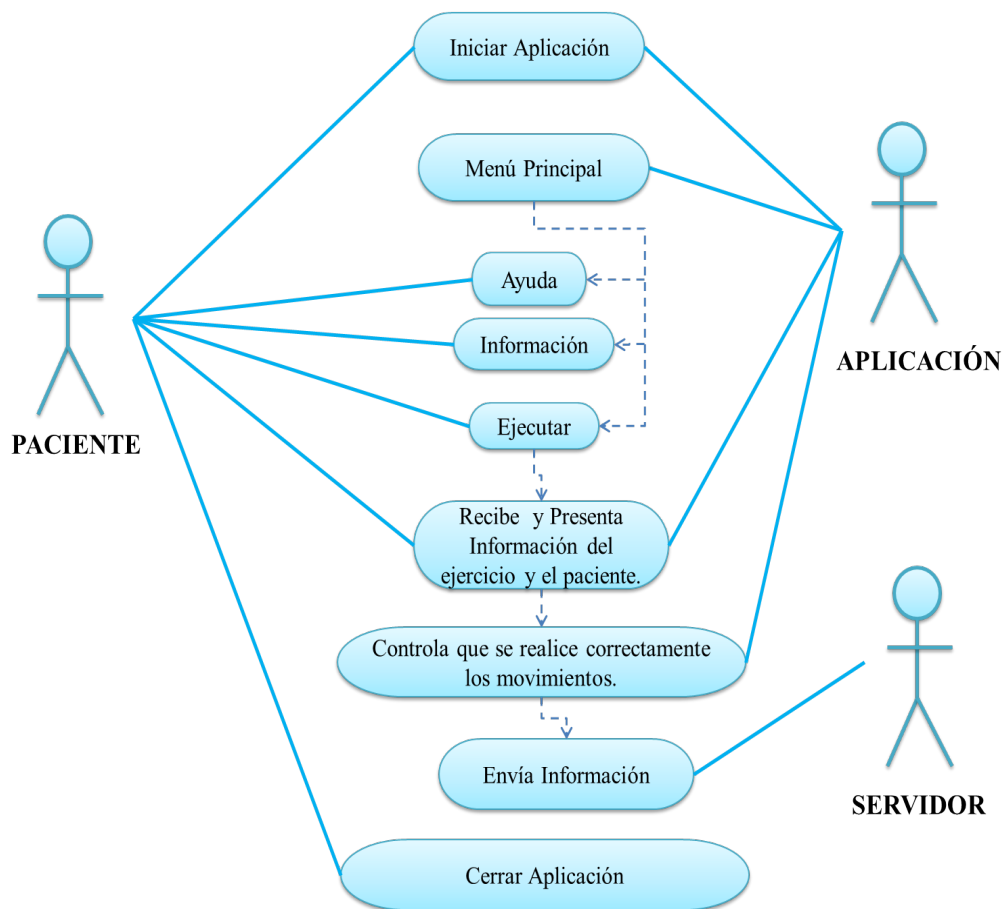


Figura 33: Diagrama de Casos de Uso.

Para describir de forma clara y precisa cada uno de los aspectos considerados en el modelo de casos de uso se ordena la información en tablas en las que constan campos como Identificador, Nombre, Descripción, Actores y otros aspectos importantes que hay que considerar para entender la función que tiene cada uno. A continuación, desde la Tabla 10 hasta la Tabla 17 se tiene la descripción de los casos de uso.



Tabla 10:

Modelo de Casos de Uso "*Iniciar Aplicación*".

<b>Identificador</b>	MCU_01
<b>Nombre</b>	Iniciar Aplicación.
<b>Actores</b>	Paciente.
<b>Descripción</b>	El usuario ejecuta la aplicación.
<b>Precondiciones</b>	Abrir la aplicación Tener conectado el Sensor Kinect
<b>Flujo</b>	El actor ejecuta la aplicación.
<b>Normal</b>	El sistema inicia la aplicación.
<b>Postcondiciones</b>	Se carga la aplicación.
<b>Requerimientos</b>	RF01

Tabla 11:

Modelo de Casos de Uso "*Menú Principal*".

<b>Identificador</b>	MCU_02
<b>Nombre</b>	Menú Principal.
<b>Actores</b>	Paciente.
<b>Descripción</b>	Permite visualizar el menú principal.
<b>Precondiciones</b>	Abrir la aplicación Tener conectado el Sensor Kinect
<b>Flujo</b>	El actor inicia la aplicación.
<b>Normal</b>	El sistema muestra el menú en pantalla.
<b>Postcondiciones</b>	Se muestra la información de la opción seleccionada
<b>Requerimientos</b>	RF02

Tabla 12:

Modelo de Casos de Uso "Ayuda".

<b>Identificador</b>	MCU_03
<b>Nombre</b>	Ayuda.
<b>Actores</b>	Paciente.
<b>Descripción</b>	Permite obtener ayuda con respecto al funcionamiento de la aplicación.
<b>Precondiciones</b>	Debe aparecer el entorno de la opción.
<b>Flujo Normal</b>	El actor selecciona la opción. El sistema muestra el entorno de la opción en pantalla.
<b>Flujo Alternativo</b>	El actor selecciona otra opción. El sistema muestra el entorno de la opción seleccionada en pantalla.
<b>Postcondiciones</b>	Salir de la opción
<b>Requerimientos</b>	RF03 y RF04

Tabla 13:

Modelo de Casos de Uso "Información".

<b>Identificador</b>	MCU_04
<b>Nombre</b>	Información.
<b>Actores</b>	Paciente.
<b>Descripción</b>	Permite tener información referente a datos propios de la aplicación, tales como versión o detalles en los que consiste la misma.
<b>Precondiciones</b>	Debe aparecer la opción

CONTINÚA 

<b>Flujo</b>	El actor selecciona la opción.
<b>Normal</b>	El sistema muestra el entorno de la opción en pantalla.
<b>Flujo Alternativo</b>	El actor selecciona otra opción. El sistema muestra el entorno de la opción seleccionada en pantalla.
<b>Postcondiciones</b>	Salir de la opción.
<b>Requerimientos</b>	RF03 y RF04

Tabla 14:

Modelo de Casos de Uso "Ejecutar".

<b>Identificador</b>	MCU_05
<b>Nombre</b>	Ejecutar.
<b>Actores</b>	Paciente.
<b>Descripción</b>	Permite interactuar con la opción seleccionada.
<b>Precondiciones</b>	Debe aparecer el menú.
<b>Flujo</b>	El actor selecciona la opción.
<b>Normal</b>	El sistema muestra el entorno de la opción en pantalla.
<b>Flujo Alternativo</b>	El actor selecciona otra opción. El sistema muestra el entorno de la opción seleccionada en pantalla.
<b>Postcondiciones</b>	Iniciar Sesión
<b>Requerimientos</b>	RF03 y RF05

Tabla 15:

Modelo de Casos de Uso "*Recibe y Presenta Información del Paciente y Ejercicio*".

<b>Identificador</b>	MCU_06
<b>Nombre</b>	Recibe y Presenta Información del Paciente y Ejercicio.
<b>Actores</b>	Paciente.
<b>Descripción</b>	Permite recibir datos del paciente al iniciar sesión y obtener datos del ejercicio al ejecutar la rutina.
<b>Precondiciones</b>	Seleccionar la opción ejecutar.
<b>Flujo Normal</b>	El actor selecciona la opción. Al iniciar sesión el sistema muestra una lista de ejercicios que se pueden seleccionar y el entorno de desarrollo de la rutina.
<b>Flujo Alternativo</b>	El actor ingresa por primera vez y debe realizar el registro de sus datos. El actor decide cancelar el proceso.
<b>Postcondiciones</b>	Validar el movimiento.
<b>Requerimientos</b>	RF06, RF07 y RF08

Tabla 16:

Modelo de Casos de Uso "*Controla que se Realice Correctamente el Movimiento*".

<b>Identificador</b>	MCU_06
<b>Nombre</b>	Controla que se Realice Correctamente el Movimiento.
<b>Actores</b>	Paciente.
<b>Descripción</b>	Permite verificar que se ejecute correctamente el movimiento para continuar la rutina.

CONTINÚA 

<b>Precondiciones</b>	Debe seleccionar el ejercicio que desea realizar
<b>Flujo</b>	El actor selecciona la opción.
<b>Normal</b>	El sistema muestra el entorno de la opción en pantalla en la cual se desarrolla la rutina y valida los movimientos.
<b>Flujo Alternativo</b>	El actor selecciona otra opción y muestra el entorno y los movimientos propios del ejercicio seleccionado. El Actor decide regresar al menú o salir de la opción.
<b>Postcondiciones</b>	Se envían los datos resultantes de la rutina a un servidor.
<b>Requerimientos</b>	RF09, RF10 y RF11

Tabla 17:

Modelo de Casos de Uso "Cerrar Aplicación".

<b>Identificador</b>	MCU_8
<b>Nombre</b>	Cerrar Aplicación.
<b>Actores</b>	Paciente/Terapeuta
<b>Descripción</b>	Permite a los actores cerrar la aplicación.
<b>Precondiciones</b>	Debe estar disponible la aplicación.
<b>Flujo</b>	El actor selecciona el botón para salir de la aplicación
<b>Normal</b>	El sistema cierra la aplicación.
<b>Postcondiciones</b>	Muestra un mensaje para asegurarse que el usuario desea salir de la aplicación.
<b>Requerimientos</b>	RF04

### 3.1.5.6 Diseño de la Interfaz Gráfica de Usuario (GUI).

Para el diseño de cada una de las interfaces que conforman el sistema se debe considerar que deben ser intuitivas, de fácil comprensión y coherente con la información mostrada, si se cumplen estas tres premisas se desarrolla aplicaciones amigables y de

fácil interacción con el usuario. A continuación se presenta los borradores del diseño de las interfaces más importantes del sistema con una breve explicación de los elementos que conforman cada una de ellas.

### Ventana "Menú Principal".

Esta es la interfaz que aparece al iniciar la aplicación (Figura 34), cuenta con botones personalizados con imágenes acordes a la función que cumplen y que permiten realizar las diferentes funcionalidades que brinda el sistema. Además, dispone de otros campos que únicamente forman parte del entorno gráfico de la aplicación. La interfaz implementada se puede apreciar más adelante en la Figura 43 (a).

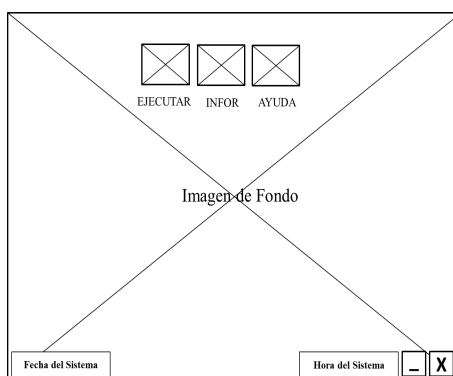


Figura 34: Diseño de la Ventana "Principal"

### Ventana "Iniciar Sesión".

El paciente debe ingresar el usuario y contraseña para poder acceder a la lista de ejercicios (Figura 35). Si el paciente no cuenta con ésta información debe ingresar en la opción registrar en la que ingresará los datos solicitados(Figura 36).

El diagrama muestra un recuadro rectangular que representa la ventana de inicio de sesión. En la parte superior izquierda hay un campo etiquado 'USUARIO:'. Debajo de él hay un campo etiquado 'CONTRASEÑA:'. A la derecha de 'CONTRASEÑA:' hay un botón etiquado 'REGISTRAR'. En la parte inferior izquierda hay un botón etiquado 'ACEPTAR'. En la parte inferior derecha hay un botón etiquado 'CANCELAR'.

Figura 35: Diseño de la Ventana "Iniciar Sesión".

Nombre:

Apellido:

Nombre:

Cédula:

Zona a Rehabilitar:

Usuario:

Contraseña:

Figura 36: Diseño de la Ventana "Registrar".

La implementación de las ventanas de login y registro se puede observar en la Figura 43 literales (b) y (c) respectivamente.

### Ventana "Lista de Ejercicios".

Consiste en un menú que muestra los ejercicios que tiene disponible el paciente para realizar la sesión de rehabilitación. Tiene botones adicionales con los que el terapeuta puede realizar acciones de agregar o eliminar los ejercicios. En la Figura 37 se observa el diseño de la lista de ejercicios, la implementación de la misma se aprecia en la Figura 43 (d).

SELECCIONE EJERCICIO

USUARIO:

Figura 37: Diseño de la Ventana "Lista de Ejercicios".

### Ventana "Opciones".

A ésta ventana se puede acceder desde el menú de ejercicios y permite calibrar el ángulo de inclinación del sensor *Kinect* mediante un slider para lograr una óptima

captura. La estructura de la ventana se puede apreciar en la Figura 38. El resultado final de la implementación de la ventana se aprecia en la Figura 44 (b).

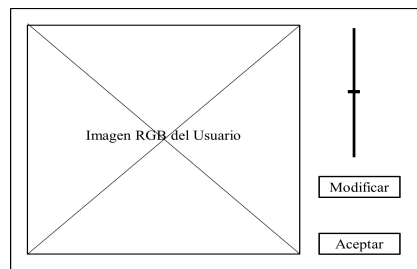


Figura 38: Diseño de la Ventana "Opciones".

### Ventana del Entorno "Realizar Ejercicio".

En un área se presenta la imagen en tiempo real del paciente y en otra más pequeña el movimiento que debe realizar. Además, se visualiza el número de repeticiones y el tiempo que transcurre con una especie de cronómetro (Figura 39). La estructura final que tiene la ventana se presenta en la Figura 44 (a).

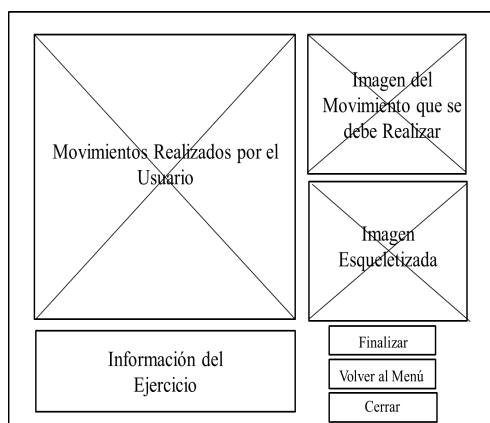


Figura 39: Diseño de la Ventana "Realizar Ejercicio".

### Ventana "Ayuda".

En la ventana de ayuda se puede visualizar un breve manual de usuario presentado en imágenes con su respectiva explicación que van cambiando al presionar un botón



(Figura 40). En la Figura 43 (f) se tiene la ventana resultante de la implementación del diseño de la opción ayuda.

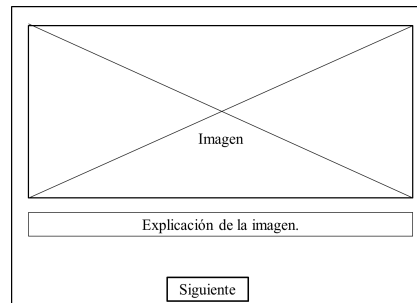


Figura 40: Diseño de la Ventana "Realizar Ejercicio".

### 3.1.6 Diseño de la Etapa del Servidor Web y Base de Datos.

La etapa del servidor web consta básicamente de dos componentes: aplicación web y base de datos, que como ya se había explicado en el diagrama de bloques de la Figura 30 de la sección 3.1.3, la aplicación web se relaciona con la interfaz de la aplicación (software del prototipo) mediante la base de datos. Por lo que en esta sección únicamente se procede a describir el diseño.

#### 3.1.6.1 Diseño de la Base de Datos.

Para que la aplicación pueda almacenar la información del paciente, del ejercicio y de la sesión se han creado tablas con campos específicos para cada parámetro almacenado. En la Figura se presenta el diseño y relación entre tablas que tiene la base de datos.

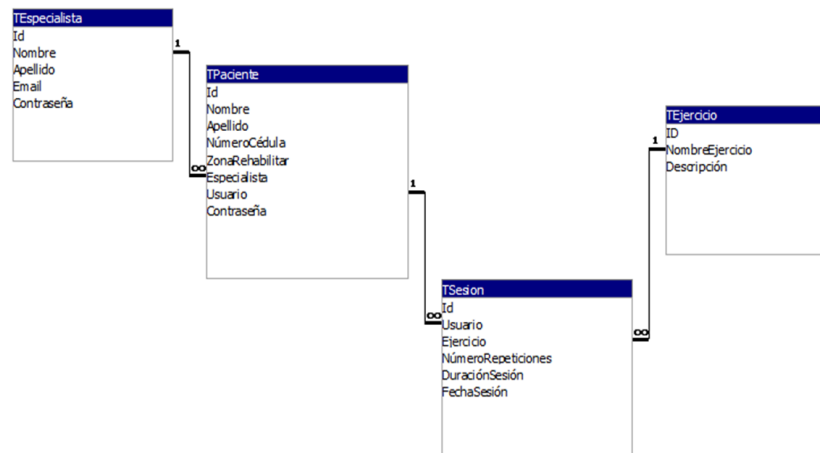


Figura 41: Modelo de la Base de Datos .

### Tabla TPaciente.

Representa a los pacientes que se han registrado para utilizar el prototipo. Consta de los campos Nombre, Apellido, Cédula, ZonaRehabilitar, Especialista, Usuario y Contraseña, los cuales serán ingresados desde la interfaz de aplicación. El identificador primario principal es el campo usuario, ya que no podrá existir un registro similar.

### Tabla TEjercicio.

En la tabla TEjercicio se almacena el nombre del ejercicio y la descripción que se tiene para el mismo. El identificador primario es el campo Id que posee.

### Tabla TEspecialista.

La tabla contiene la información que el especialista ingresa desde la aplicación Web para registrarse en el sistema, tiene los campos Nombre, Apellido, Email y Contraseña.

### Tabla TSesion.

En la tabla TSesion se almacenan los datos proporcionados por la aplicación cada

vez que un paciente acceda a la misma. Cuenta con los campos Numero Sesión que es el identificador primario de la tabla y que se asigna automáticamente cuando se produce un nuevo registro, el campo Usuario que permite relacionarse con la tabla TPaciente, Nombre Ejercicio con el que se puede acceder a la tabla TEjercicio y los tres últimos campos que se registran cuando un paciente ha finalizado la rutina de ejercicios.

### 3.1.6.2 Diseño de la Interfaz de la Aplicación Web.

La aplicación web constará de dos ventanas para ingresar y presentar la información. La primera ventana permitirá ingresar a consultar los datos de la sesión realizada por el paciente y el registro del especialista en el sistema. En la segunda ventana, el terapeuta podrá seleccionar el usuario del que desea visualizar los datos, los mismos que serán presentados en una tabla. En la Figura 42 se tiene la estructura gráfica de las ventanas.

(a) Inicio de Sesión

(b) Información de Paciente

Figura 42: Diseño de la Interfaz de la Aplicación Web.

El resultado final de la implementación de la aplicación web se presenta en la Figura 72 y en la Figura 73, las cuales corresponden a los diseños (a) y (b) de la Figura 42.

## 3.2 Implementación del Prototipo Auxiliar para Fisioterapia.

Una vez que se ha realizado el diseño del sistema y definiendo los requisitos que debe cumplir, se procede a implementar las funcionalidades que hacen que la aplicación se ejecute adecuadamente. A continuación se explican los aspectos más relevantes que se han considerado para la implementación de la interfaz del prototipo auxiliar en el desarrollo de rutinas de rehabilitación.

### 3.2.1 Herramientas y Recursos

Para el desarrollo del proyecto se va a trabajar con las siguientes herramientas y recursos:

#### Hardware

- PC con procesador Intel Core i7 2.40 GHz, 12 GB de RAM(de los cuales se requiere 2 GB)y una tarjeta de vídeo dedicado de 4 GB que soporta Direct X, que son algunos de los requisitos para trabajar en aplicaciones con *Kinect*.
- Dispositivo *Kinect* para *Xbox 360*.
- Cable de corriente con puerto USB para *Kinect*.

#### Software

- Windows 8.1
- Microsoft .NET Framework 4.5.1 (Incluido con el Visual Studio Profesional 2013)
- Kit de Desarrollo de Software (SDK) para *Kinect* última versión 1.7.

La elección del lenguaje de programación viene condicionada en base al SDK de *Kinect* que soporta únicamente C++, C# y Visual Basic en Microsoft Visual Studio.

Para el desarrollo del trabajo se ha seleccionado el lenguaje **C Sharp (C#)**, que consiste en una programación orientada a objetos. La aplicación de C# que se utiliza es **Windows Presentation Foundation (WPF)**, la cual ofrece un amplia gama de herramientas gráficas para desarrollar interfaces visualmente atractivas. Al permitir la fácil integración de elementos de vídeo, animación, audio e imágenes se convierte en la opción más adecuada para el desarrollo de aplicaciones con *Kinect*.

### 3.2.2 Interfaz de Aplicación.

En esta sección se presenta la interfaz de la aplicación desarrollada en WPF que se implemento con ayuda de los controladores que dispone en la Caja de Herramientas y que se basan en lenguaje XAML que es muy utilizado para la presentación visual de una aplicación. Los más necesarios fueron los siguientes:

Tabla 18:

Controles de las Ventanas de la Aplicación.

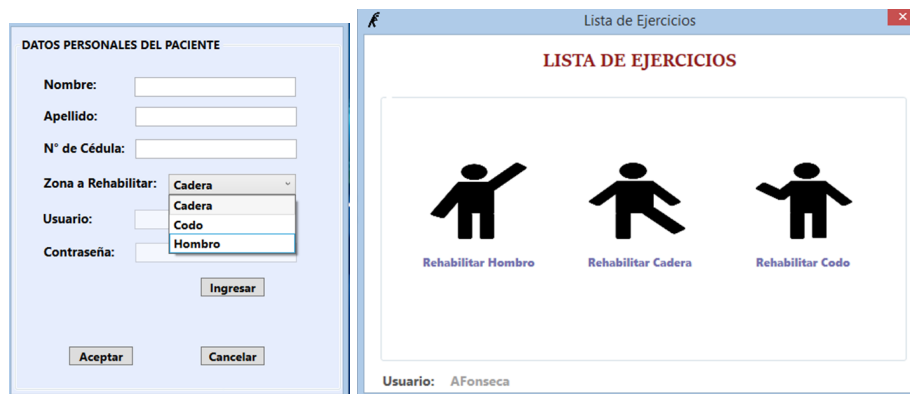
Control	Descripción
Botón	Permite realizar las acciones al hacer click sobre el elemento.
Label	Permite mostrar texto en la ventana.
TextBox	Muestra texto y permite a los usuarios ingresar datos.
Image	Permite presentar imagenes.
PasswordBox	Permite ingresar texto pero como una contraseña.
Canvas	Sirve de lienzo para dibujar líneas, elipses, puntos, etc.
GroupBox	Contenedor que permite agrupar controles.
Grid	Define un área de cuadrícula flexible que consta de columnas y filas.
Slider	Permite obtener un valor numérico al arrastrar el cursor.
ComboBox	Es una lista desplegable.

Las interfaces elaboradas se pueden apreciar en la Figura 43.



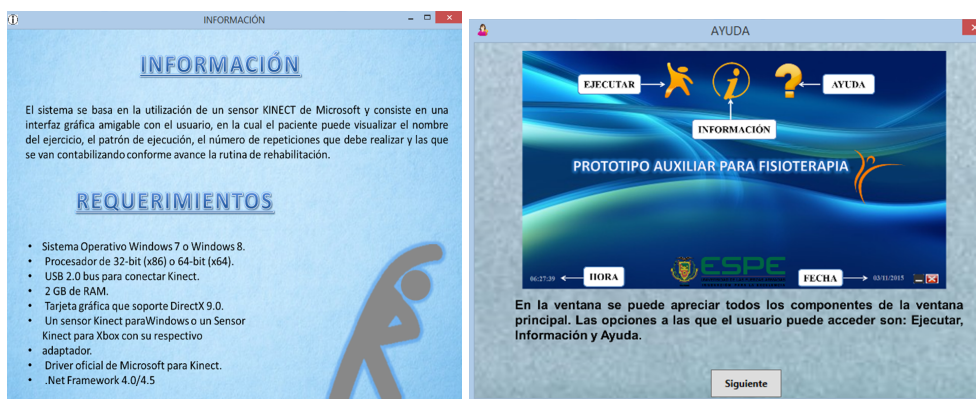
(a) Inicio

(b) Iniciar Sesión



(c) Registrar

(d) Lista de Ejercicios

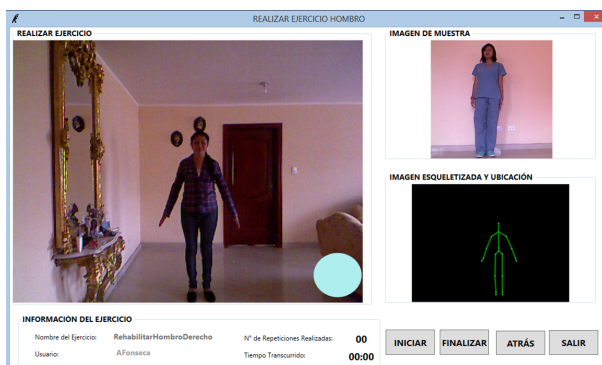


(e) Información

(f) Ayuda

Figura 43: Interfaz de Aplicación.

Las interfaces en las que se presenta la información que se obtiene del sensor *Kinect* se implementaron de acuerdo al diseño, como se puede observar en la Figura 44.



(a) Realizar Ejercicio



(b) Opciones

Figura 44: Interfaz de Aplicación de las Ventanas "Realizar Ejercicio" y "Opciones".

### 3.2.3 Flujo de Datos del Sensor Kinect.

Hay que tener claro que cuando un sensor *Kinect* está activado puede capturar datos de color, profundidad y audio, además procesando y combinando los datos provenientes del sensor RGB y de profundidad se puede acceder a los datos de esqueletos. Por lo tanto, el sensor *Kinect* lo que proporciona son flujos de datos, a los cuales se puede tener acceso y control mediante la programación con la ayuda de las librerías propias del *Kinect* que al inicializarse requieren tener en la aplicación los siguientes aspectos.

- Identificar los flujos de datos que se necesitan.

- Abrir o habilitar el flujo de datos.
- Asignar previamente buffers para contener los datos del sensor.
- Obtener los datos nuevos para cada flujo de cada frame.
- Vaciar el buffer de modo que el tiempo de ejecución puede llenarlo con el siguiente fotograma.

Para activar el flujo de datos del sensor *Kinect* se debe crear un objeto de la clase *KinectSensor*, en esta clase se especifica parámetros como el número de frames por segundo que se desea mostrar y la información de profundidad, de color o skeleton que se desea obtener. Es la más importante ya que permite establecer si el sensor se encuentra conectado y listo para ser usado, desde esta clase también se realiza la llamada al método `Start()` para inicializar la captura de información proveniente del sensor. Además, hay que tener en cuenta que se puede tener más de un sensor conectado a la PC por lo que se debe especificar dentro del arreglo de sensores el que se va a utilizar. En la Figura 45 se puede apreciar las sentencias que crean el objeto, habilitan el sensor *Kinect* e identifican la posición del sensor que se va a utilizar dentro del arreglo de sensores.

```
// Objeto de la Clase KinectSensor
KinectSensor miSensorKinect;
//Elige el primer sensor que encuentra
miSensorKinect = KinectSensor.KinectSensors.FirstOrDefault();
//Inicia la captura de datos del sensor Kinect
miSensorKinect.Start();
```

Figura 45: Sentencias para Iniciar el *Kinect*.

El diagrama de flujo que describe el procedimiento de inicio de la transmisión de datos de *Kinect* es el que se tienen en la Figura 46.



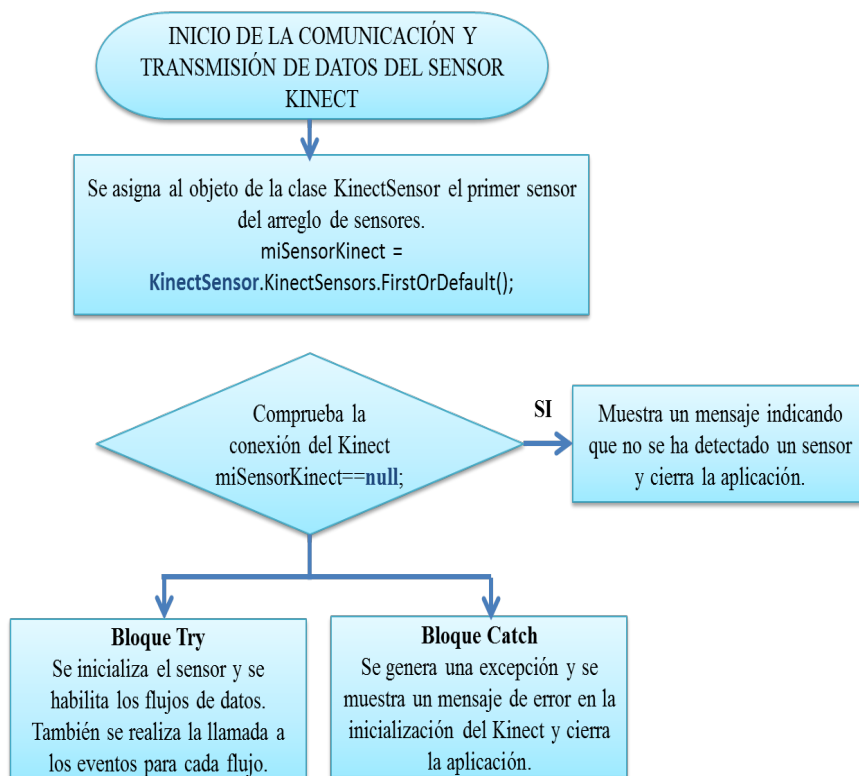


Figura 46: Diagrama de Flujo del Algoritmo utilizado para Iniciar la Transmisión de Datos y Habilitar los Flujos.

### 3.2.3.1 Flujo de Datos de Color.

Los datos de color están disponibles en diferentes formatos y resoluciones. Para habilitar el flujo de color se utiliza las sentencias que se tiene en la Figura 47 , en la misma figura se puede apreciar también la llamada al evento que realiza todo el proceso de captura de los datos y creación de la imagen proveniente del sensor *Kinect* que se mostrará en pantalla.

```

//Para habilitar el flujo de Color
miSensorKinect.ColorStream.Enable();
//Evento del proceso de flujo de Imágenes de Color
miSensorKinect.ColorFrameReady += miSensorKinect_ColorFrameReady;
  
```

Figura 47: Sentencias para Habilitar el Flujo de Color del *Kinect*.

Únicamente se puede tener un flujo de imágenes de color por sensor, por lo que

se debe restablecer el flujo de imágenes de color existente. Este proceso es el que se realiza en el EventHandler del flujo de color, para lo cual se utilizan las funciones que se presentan en la Tabla 19 y cuyo diagrama de flujo del algoritmo se presenta en la Figura 48.

Tabla 19:

Funciones para Presentar la Imagen de Color.

Función	Descripción.
<code>ColorImageFrame</code>	Proporciona acceso a los datos de dimensiones, de formato y de píxeles para una trama.
<code>WriteableBitmap</code>	Proporciona un conjunto de píxeles con un tamaño y una resolución determinados, en el que se puede grabar o actualizar.

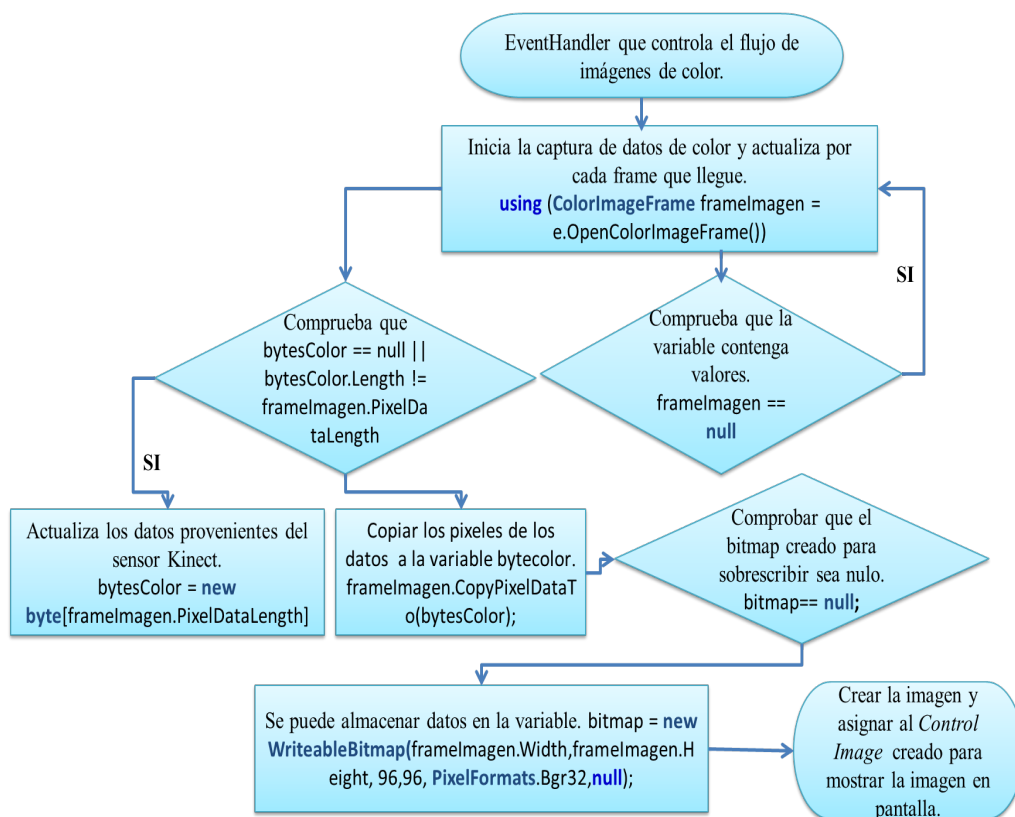


Figura 48: Diagrama de Flujo del Algoritmo utilizado para la Imagen de Color.

De la implementación del procesamiento de los datos que proporciona el sensor al habilitar el flujo de color se obtienen la imagen que se presenta en la Figura 49.

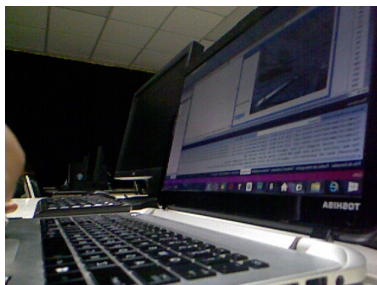


Figura 49: Imagen de Color.

### 3.2.3.2 Flujo de Datos de Esqueletos.

Al habilitar esta función se envía el flujo de frames constantes con las posiciones 3D de las articulaciones mediante la función **SkeletonTrackingstate** que permite almacenar, en variables tipo double, las coordenadas en (x,y,z). Las sentencias que se utilizan para definir las coordenadas son las que se aprecian en el ejemplo de la Figura 51, considerando cada articulación reconocida por el sensor con el nombre en inglés de la Figura 50.

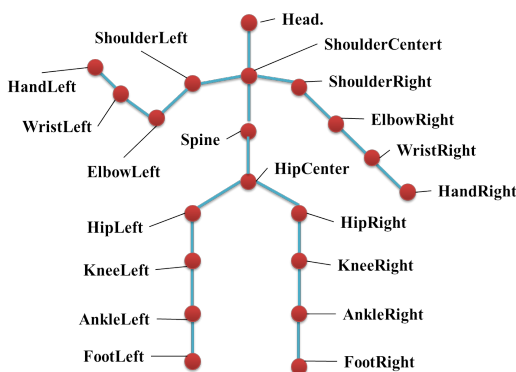


Figura 50: Articulaciones Reconocidas por *Kinect*.

```

Codigo.X = skeleton.Joints[JointType.ElbowRight].Position.X;
Codigo.Y = skeleton.Joints[JointType.ElbowRight].Position.Y;
Codigo.Z = skeleton.Joints[JointType.ElbowRight].Position.Z;

```

Figura 51: Sentencias para Definir las Coordenadas de los Joints.

La información de los joints (articulaciones) permiten realizar procesos que dependen de los requerimientos de la aplicación, al trabajar con las posiciones se puede establecer una interfaz entre el usuario y la aplicación.

En la Tabla 20 se aprecia la descripción de las funciones necesarias para desarrollar el proceso que controla el flujo de datos del esqueleto reconocido por el sensor. Por otro lado, en la Figura 52 se tiene el flujo de datos del proceso.

Tabla 20:

Funciones para Controlar el Flujo de Esqueletos.

<b>Función</b>	<b>Descripción.</b>
<a href="#">Skeleton</a>	Proporciona toda la información para la creación del esqueleto e identificación del usuario.
<a href="#">SkeletonFrame</a>	Permite inicializar la captura del frame de esqueletos. Contienen la información de posición y articulaciones que es reconocida por el sensor.
<a href="#">SkeletonTracking State</a>	Permite realizar un seguimiento del esqueleto e identifica las articulaciones. Entrega los posibles estados de Skeleton.

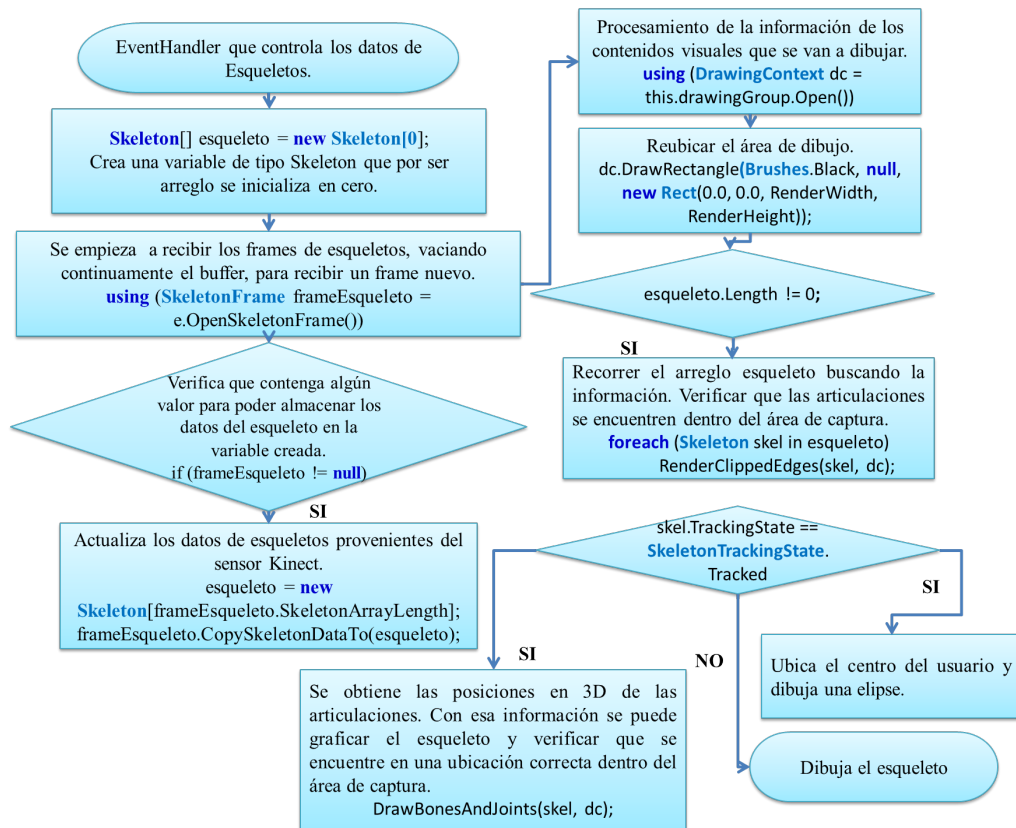


Figura 52: Diagrama de Flujo del Algoritmo Utilizado para el Control de los Datos de Esqueletos.

La imagen de la Figura 53 resulta de la habilitación de la opción de trackeo y del procesamiento de las coordenadas de las articulaciones del cuerpo humano.

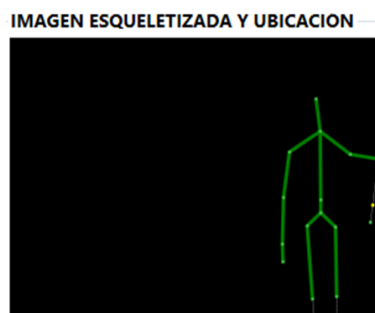


Figura 53: Imagen Esqueletizada.

### 3.2.4 Reconocimiento de Movimientos.

Para el reconocimiento de los movimientos lo primero que se debe hacer es definir el movimiento que se desea reconocer, para lo cual se ha utilizado un algoritmo vectorial que consiste en calcular vectores en base a las coordenadas de las articulaciones detectadas por el sensor y posteriormente encontrar el ángulo entre dos vectores. La comparación se realiza tomando en cuenta los ángulos obtenidos y un valor de referencia adecuado para dar como válido el movimiento ejecutado por el usuario.

Se definen cuatro coordenadas de referencia, que vienen a ser dos vectores de referencia horizontal y vertical (Figura 54), debido que se tenía que estabilizar la variación de las articulaciones y del ángulo que de ellas se obtiene para lograr un reconocimiento óptimo. El vector de referencia en el que intervienen las articulaciones de la cadera y la espina se utiliza para calcular el ángulo vertical (AVR), mientras que el otro vector de referencia se utiliza para definir el ángulo horizontal (AHR).

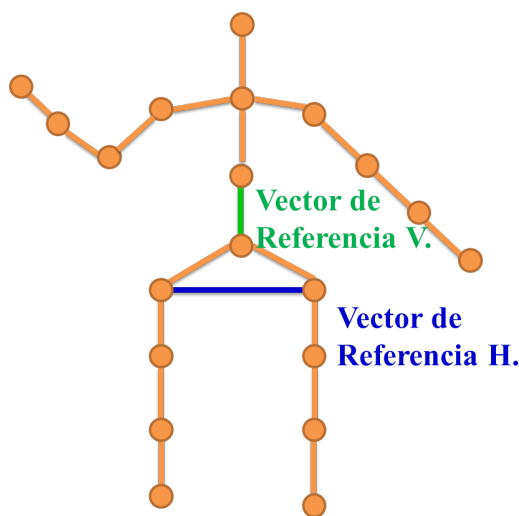


Figura 54: Vectores de Referencia.

Los vectores de referencia corresponden a uno de los vectores que intervienen en el cálculo del ángulo, para el segundo vector se considera las articulaciones que están involucradas en el movimiento. Por ejemplo para el ejercicio de hombro se debe mover el brazo derecho por lo que el segundo vector corresponde a las articulaciones

del hombro y codo derecho, en la Figura 55 se puede apreciar de forma gráfica los vectores y los ángulo AVR - AHR que se consideran para el ejercicio de hombro.

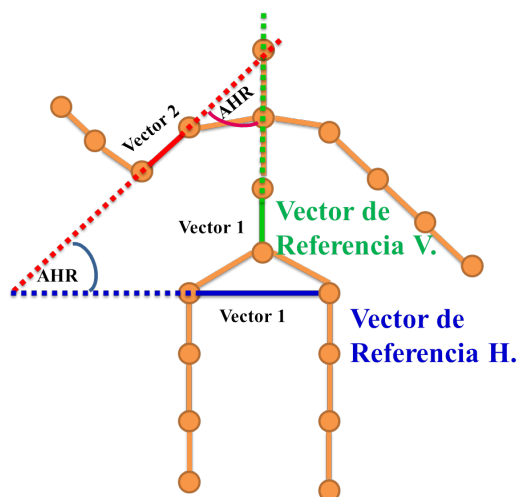


Figura 55: Ángulo AVR y AVH para el Ejercicio Hombro.

Nota.- Para los movimientos que implican los ejercicio de cadera y codo se debe analizar la extremidad el cuerpo que va realizar el movimiento para definir las articulaciones que formarán el segundo vector, considerando un procedimiento similar al realizado para el ejercicio de hombro.

La implementación del movimiento que se desea detectar se realiza en base al algoritmo que se detalla en el diagrama de flujo de la Figura 56.

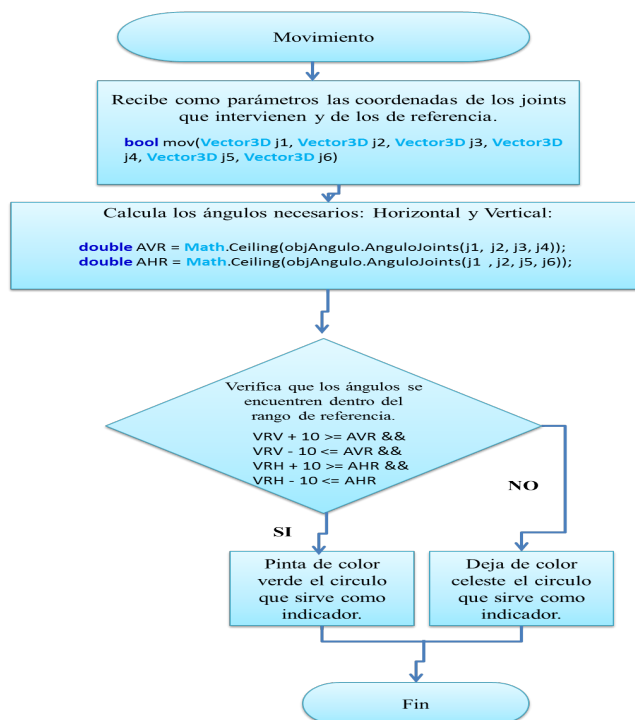


Figura 56: Diagrama de Flujo del Algoritmo Utilizado para Definir el Movimiento a Detectar.

Una vez que se ha definido el movimiento se realiza un detector de posturas, en el que se lleva la cuenta de las posturas que se encuentra en proceso de detección, ya que una postura debe permanecer durante un cierto intervalo de tiempo para que pueda considerarse como detectada. Para ello se define un enumerador que facilitará hacer referencia al movimiento creado, el mismo que tiene el código que se presenta en la Figura 57.

```

const int PostureDetectionNumberC = 10;
int accumulatorC = 0;
PostureC postureInDetectionC = PostureC.None;
PostureC previousPostureC = PostureC.None;
public enum PostureC
{
    None,
    Movimiento
}
  
```

Figura 57: Creación del Enumerador.



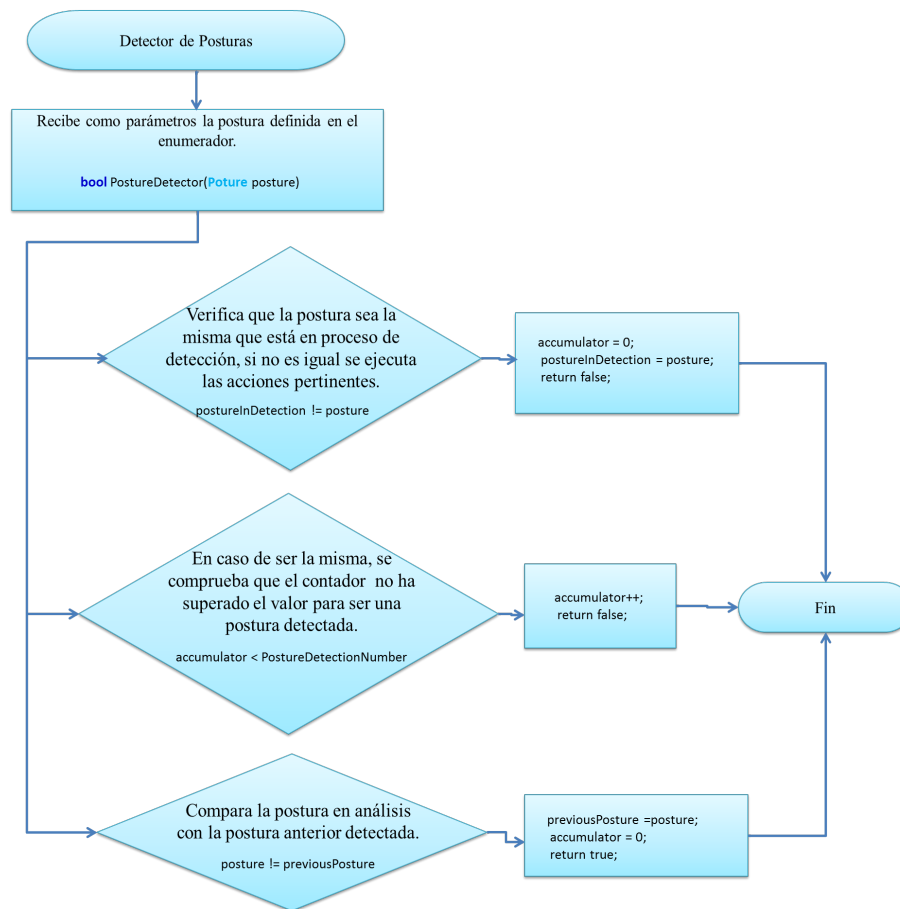


Figura 58: Diagrama de Flujo del Detector de Posturas.

En el detector de posturas se controla la postura que se está ejecutando por parte del paciente, la postura en detección, la postura anterior y un contador que lleva la cuenta del número de veces que se ha analizado el movimiento para definir si es válido. La lógica utilizada para implementar el método es la que se presenta en el diagrama de flujo de la Figura 58.

La llamada a la función se realiza después de haber almacenado los joints en el vector creado con ese propósito, ya que es necesario enviar la información de coordenadas para realizar las operaciones, detectar los movimientos y así incrementar un contador que controla el número de repeticiones o el número del movimiento que se está ejecutando, en el caso de que el ejercicio conste de más de un movimiento. El

diagrama de flujo del proceso se puede ver en la Figura 59.

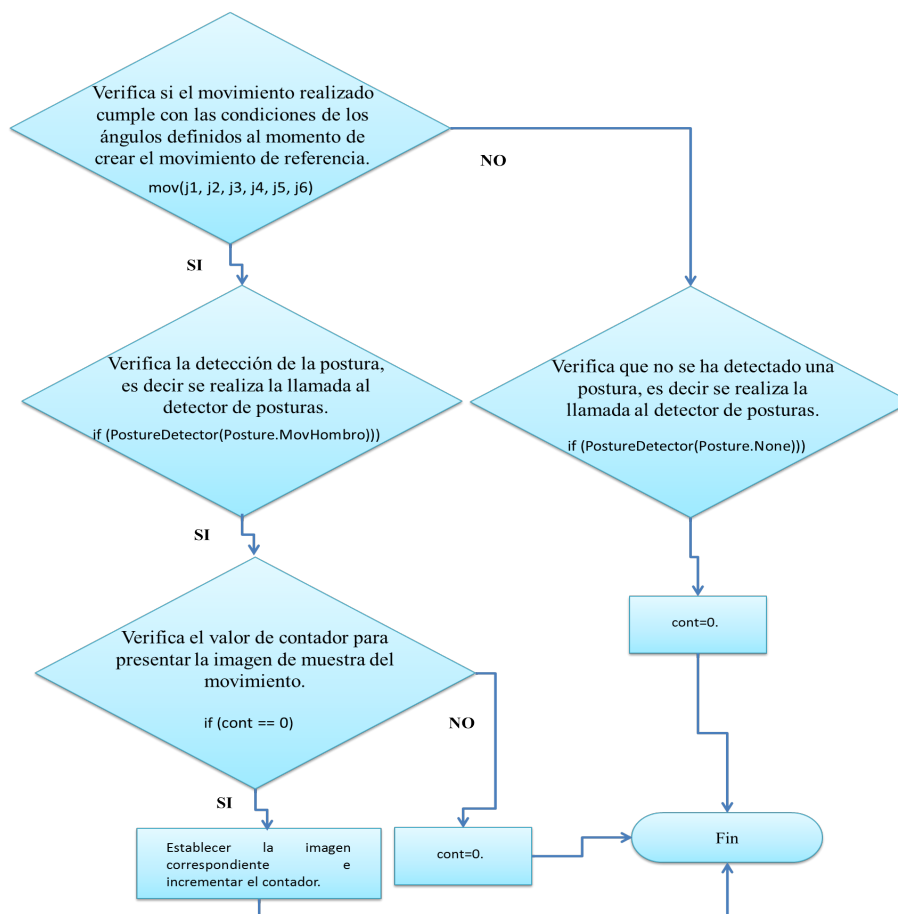


Figura 59: Diagrama de Flujo de la Llamada a las Funciones de Detección de Movimientos.

Los procesos presentados en los diagramas de las Figuras ( Figura 56, Figura 57 y Figura 58) se repite por cada movimiento que se desea detectar y la llamada se realiza de la misma manera que se presenta en el diagrama de la Figura 59, con los procesos que se ejecutan adaptados a las necesidades de la aplicación, en éste caso presentar la imagen e incrementar un contador para llevar la cuenta de las repeticiones correctas del ejercicio.

### 3.2.5 Reconocimiento de Voz

El sensor *Kinect*, gracias a que dispone de un arreglo de micrófonos, posibilita el reconocimiento de voz. Para lo que se debe utilizar las librerías de reconocimiento de

audio y la de formato de audio, las cuales se obtienen e instalan igual que el SDK. A los métodos que ofrecen estas librerías se puede acceder mediante la creación de un objeto de tipo **SpeechRecognitionEngine**, que es la clase que permite el reconocimiento de voz. El diagrama de Flujo que detalla el procedimiento se puede apreciar en la Figura 60.

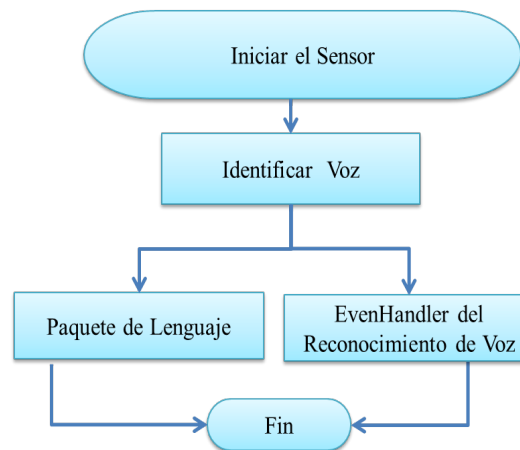


Figura 60: Diagrama de Flujo del Reconocimiento de Voz.

En el bloque en el que identifica la voz se definen las frases que se va a reconocer, el lenguaje que se va a utilizar, el evento que ejecuta las opciones y se realiza la configuración de parámetros como el formato de codificación, el número de muestras por segundo, bits por muestra, número de canal, promedio de bytes por segundo y los bytes que se van a transmitir. Los valores utilizados en el desarrollo del proyecto se tienen en la Figura 61 , en la que se puede apreciar también que se tiene un reconocimiento en modo múltiple.

```

//                                     (formato de codificacion PCM, muestras por Segundo,
//                                     bits por Muestra, número de canal, promedio Bytes por Segundo,
//                                     alineacion del Bloque, bytes [] Formato especifico de los datos)
motorLenguaje.SetInputToAudioStream(miSensorKienct.AudioSource.Start(),
                                   new SpeechAudioFormatInfo(EncodingFormat.Pcm, 16000, 16, 1, 32000, 2, null));
//Sincronizar el reconocimiento de modo multiple
motorLenguaje.RecognizeAsync(RecognizeMode.Multiple);
  
```

Figura 61: Sentencias para la Configuración del Reconocimiento de Voz.

En el eventhandler que se crea desde el identificador de voz se establecen las acciones que se ejecutan con el comando establecido mediante una sentencia de control switch, además se define un rango de tolerancia de error para la interpretación de la palabra o frase. Mientras que en el bloque correspondiente al paquete de lenguaje se configura el idioma que se va a identificar.

### 3.2.6 Descripción de las Clases y Métodos Implementados.

En esta sección se hace una descripción breve de la utilidad que tienen y se presenta la lógica de programación que se utiliza para su implementación.

#### 3.2.6.1 Clase Datos *Kinect*.

En la clase *Datos Kinect* se realiza toda las operaciones en las que interviene la información del sensor *Kinect*. Los métodos implementados son los siguientes:

- **Método Verificar el Área de Captura.**

Para lograr esta funcionalidad se utiliza el método *ClippedEdges* propio de la librería de *Kinect*. Esta propiedad del SDK permite almacenar si el rango de captura no identifica ciertas articulaciones y compara con un identificador que se denomina **FrameEdges** que viene definido en el mismo SDK y que contiene las distintas opciones que puede tener el *ClippedEdges*. En la Figura 62 se puede observar las opciones que tiene la propiedad.

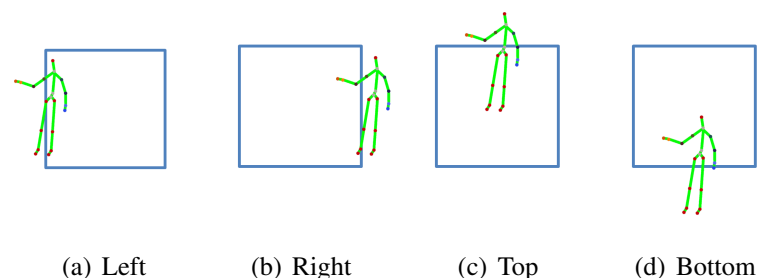


Figura 62: Valores que Determinan la Calidad de Captura.

Con el método *Verificar el Área de Captura* se puede identificar en pantalla si el paciente se encuentra ubicado en el área de captura adecuada, ya que se puede visualizar una imagen con fondo de color negro en la cual se pintan rectángulos de color rojo en el espacio de captura que el sensor *Kinect* no puede identificar correctamente. La lógica de programación que se aplicó fue la que se observa en la Figura 63.

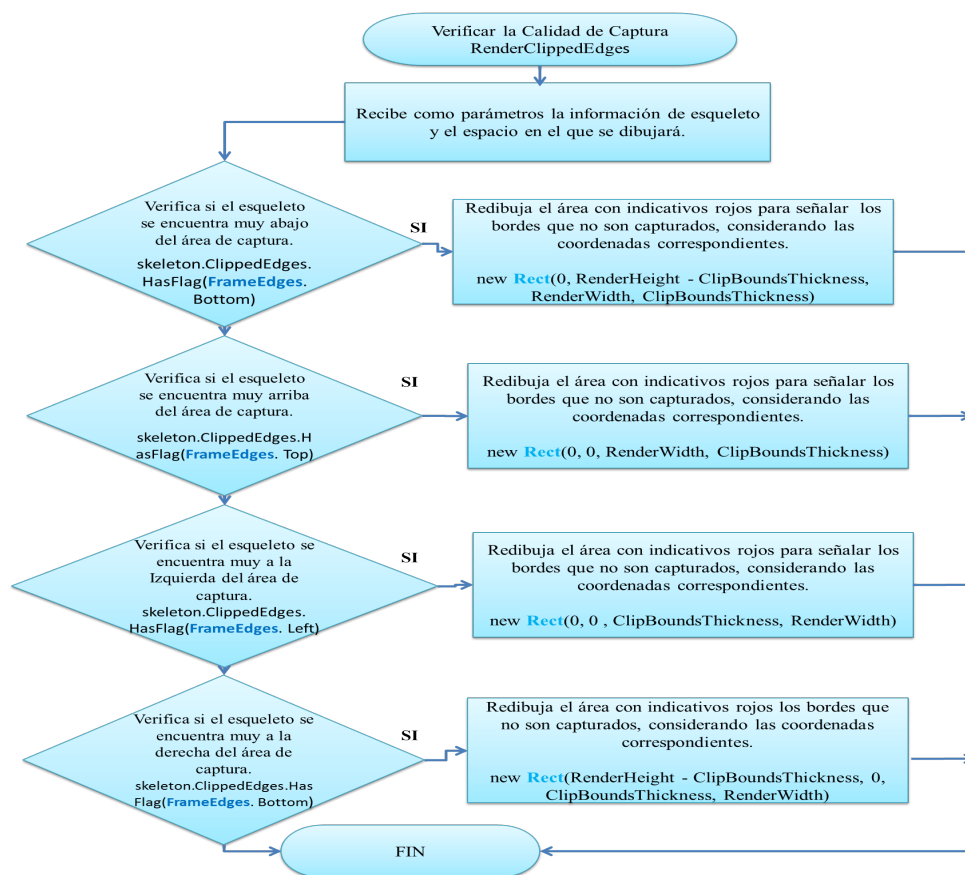


Figura 63: Diagrama de Flujo del Algoritmo utilizado para el Verificar el Área de Captura.

- **Método Dibujar Huesos.**

Se realiza el *tracking* del esqueleto reconocido por el sensor y permite obtener la información de coordenadas de las articulaciones. Con esa información se puede crear líneas o rectángulos que representen los huesos del cuerpo humano, el *Método Dibujar Huesos* se implementa para realizar ésta operación. Además,

con la información que se obtiene de las articulaciones se realiza la llamada a las funciones que permiten el reconocimiento de movimientos. La lógica que se aplicó fue la que se describe en la Figura 64.

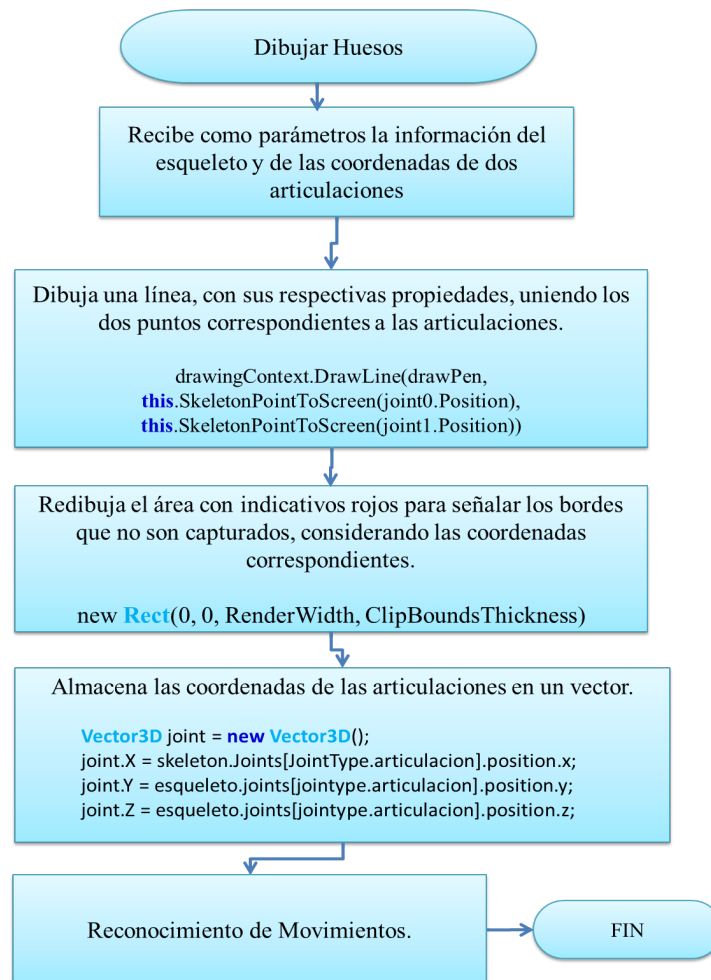


Figura 64: Diagrama de Flujo del Algoritmo utilizado para el Dibujar Huesos.

### 3.2.6.2 Clase Ángulo.

Se calcula el ángulo, ya que es la información que se va a comparar para determinar si el movimiento está ejecutado correctamente. Éste método es el más conveniente, debido a que es independiente de la estatura del usuario y del entorno en el que se

desarrolle la sesión de rehabilitación.

Para determinar el valor del ángulo se hace uso de vectores, los cuales se obtendrán con las coordenadas de las articulaciones. Las operaciones matemáticas con las que se obtiene los vectores y el ángulo se realizan en el método *Calcular el Ángulo entre Articulaciones*, el mismo que tiene la siguiente lógica mostrada en la Figura 67.

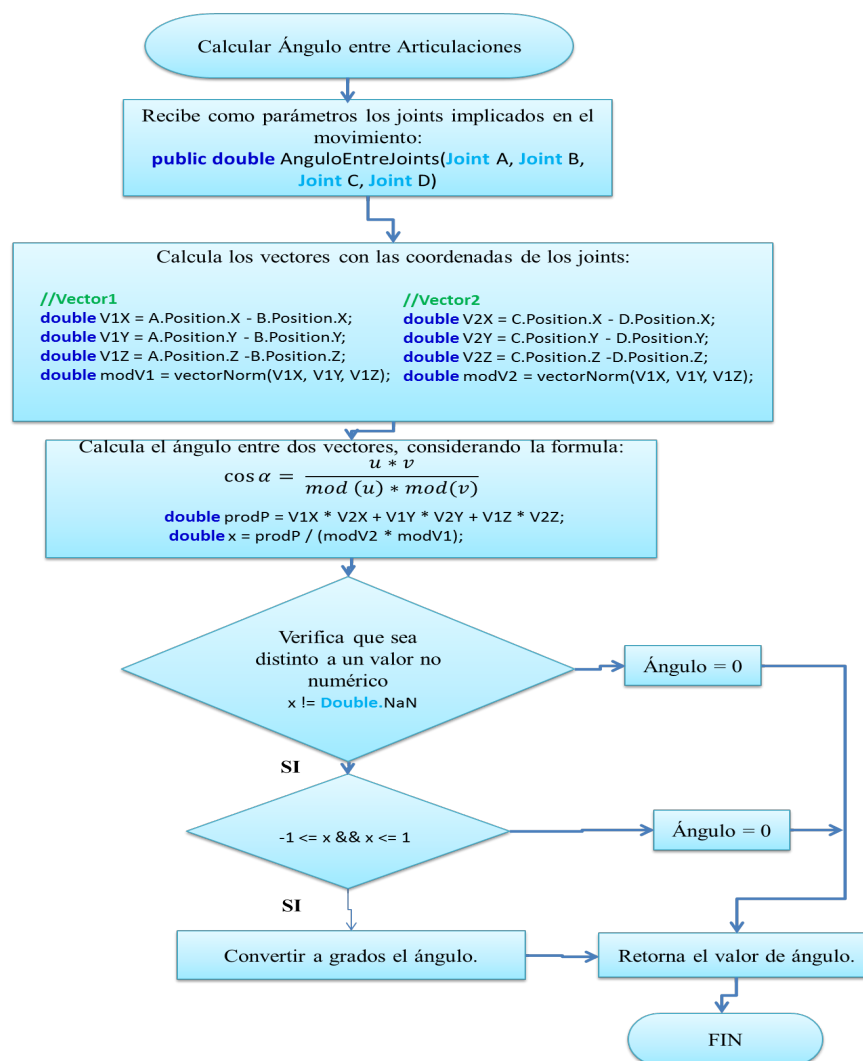


Figura 65: Diagrama de Flujo del Algoritmo Utilizado para Calcular Ángulo.

### 3.2.6.3 Clase Cronómetro.

Permite obtener el tiempo estimado que demoró el paciente en realizar la sesión. La clase y funciones que se utilizan para la implementación son los que se detallan en la Tabla 21.

Tiene dos constructores uno que no recibe argumentos y en el que se inicializan los parámetros propios de la clase y otro que recibe como argumento una caja de texto en la que se presenta los valores del cronómetro.

En el constructor de la clase que recibe argumentos se inicializa los parámetros que se van a utilizar y tiene métodos que se describen a continuación de la Tabla 21.

Tabla 21:

Clases y Funciones para Controlar el Cronómetro.

Clase/Función	Descripción.
<code>DispatcherTimer</code>	Un contador de tiempo que procesa un intervalo de tiempo específico y a una prioridad <code>DispatcherTimer</code> específica.
<code>DispatcherTimer.Interval</code>	Obtiene o establece el período de tiempo
<code>Start()</code>	Inicia el <code>DispatcherTimer</code> .
<code>Stop()</code>	Detiene el <code>DispatcherTimer</code> .

En el mismo constructor se declara e inicia el temporizador a través de su propiedad *interval*, recibe como parámetros el `textblock` en el cual se muestran los resultados y el `textbox` en el que se tiene el valor máximo que tendrá el cronómetro. También se realiza la llamada al evento `tick` que será el que realice el cálculo de los minutos y segundo cuando se genere un cambio en el *DispatcherTimer*. Las sentencias utilizadas son las que se tiene en la Figura 66.



```

Temporizador = new System.Windows.Threading.DispatcherTimer();
Temporizador.Interval = new TimeSpan(0, 0, 0, 1);
Temporizador.Tick += Temporizador_Tick;
Temporizador.IsEnabled = true;

```

Figura 66: Setencias para el Método Iniciar el Cronometro.

Los métodos implementados en la clase Cronómetro son los que se tiene en la Tabla 22.

Tabla 22:

Métodos de la Clase Cronómetro.

Tipo	Método	Descripción.
void	<b>Temporizador Tick (object sender, EventArgs e)</b>	Es el evento tick del DispatcherTimer, se establece el valor de los minutos y segundo cada vez que el valor i se incremente.
void	<b>Start ()</b>	Utiliza la propiedad DispatcherTimer que tiene el objeto creado para iniciar la cuenta del cronómetro. El método será llamado desde el botón "Iniciar" de la interfaz "Realizar Ejercicio".
void	<b>Stop()</b>	Utiliza la propiedad DispatcherTimer que tiene el objeto creado para detener la cuenta del cronómetro. El método será llamado desde el botón "Finalizar" de la interfaz "Realizar Ejercicio".

#### 3.2.6.4 Clase Persona.

Con la ayuda de esta clase se puede crear objetos que permiten acceder a los métodos y almacenar la información de un paciente. Tiene como atributos el Nombre,

Apellido, Cédula, Zona a Rehabilitar, Usuario y Contraseña, estas variables son encapsuladas definiéndose para cada una las propiedades get y set que son a su vez sus métodos.

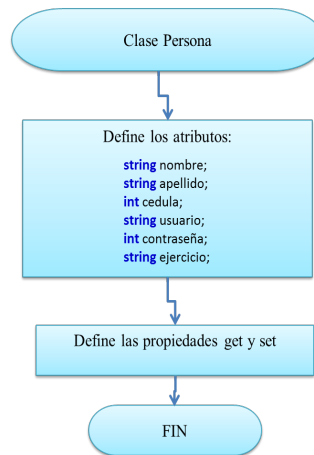


Figura 67: Diagrama de flujo del Algoritmo Utilizado en la Clase Persona.

### 3.2.6.5 Clase Vector3D.

Se puede crear objetos que permiten acceder a los métodos y almacenar la información de las coordenadas de las articulaciones. Tiene como atributos las variables x, y, z que representan las coordenadas 3D, estas variables son encapsuladas definiéndose para cada una las propiedades get y set que son a su vez sus métodos. El diagrama de flujo de la lógica implementada se muestra en la Figura 68.

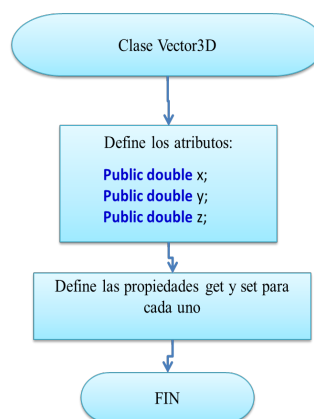


Figura 68: Diagrama de flujo del Algoritmo Utilizado en la Clase Vector3D.

### 3.2.6.6 Clase DatosSesión.

Al igual que las dos clases anteriores permite crear objetos de la clase Datos Sesión para poder acceder a sus métodos y almacenar la información de la sesión de rehabilitación realizada por el paciente. Tiene como atributos el usuario del paciente, el nombre del ejercicio, el número de repeticiones que se cumplió en la sesión, la duración y la fecha de ejecución, las variables son encapsuladas para obtener las propiedades get y set. El diagrama de flujo es presentado en la Figura 35.

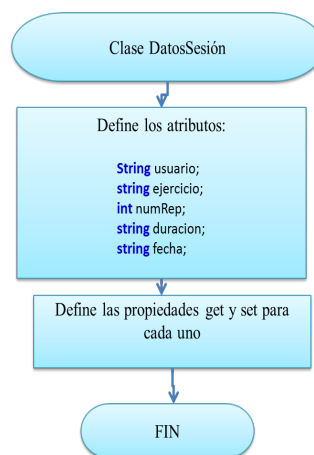


Figura 69: Diagrama de flujo del Algoritmo Utilizado en la Clase DatosSesión.

### 3.2.6.7 Clase EnviarInfSesion.

En esta clase se realiza la llamada a los métodos de la clase ConexionBD en los que se inserta la información de la sesión de rehabilitación, se hace consultas a la base de datos para obtener el nombre del especialista en base al usuario del paciente para conocer la dirección de correo electrónico al que se va a enviar la información. El diagrama de flujo de la clase es el que se presenta en la Figura 70.

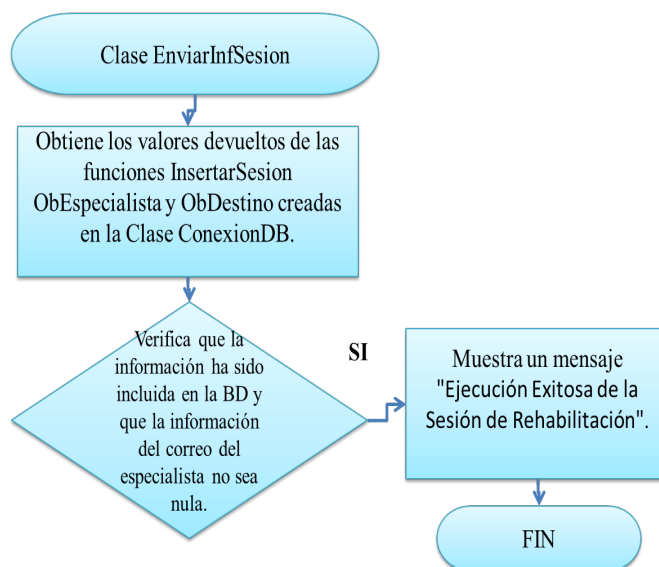


Figura 70: Diagrama de flujo del Algoritmo Utilizado en la Clase EnviarInfSesion.

### 3.2.6.8 Clase ConexionBD.

Permite conectar la base de datos creada para el prototipo, en esta misma clase se definen los métodos de ingreso y consulta de información. Los métodos implementados en la clase se presentan en la Tabla 23.

Tabla 23:

Métodos de la Clase ConectarBD.

Tipo	Método	Descripción.
void	<b>Conectar()</b>	Establece la cadena de conexión y activa la comunicación con la BD a través de la propiedad Open de los métodos propios de la conexión a BD mysql.
void	<b>Desconectar ()</b>	Cierra la comunicación mediante la propiedad Close.

CONTINÚA →

<b>Tipo</b>	<b>Método</b>	<b>Descripción.</b>
void	<b>Insertar (Persona p)</b>	Llama al método Conectar() y mediante comandos sql realiza el registro de la información que recibe como parámetro en la BD.
Persona	<b>validarContraseña (string usuario, int contraseña)</b>	Compara los datos ingresados en el formulario con los almacenados en BD. Si coinciden retorna la información de ese usuario a una variable de tipo Persona.
bool	<b>InsertarSesion (DatosSesion Info)</b>	Mediante un comando sql ingresa la información de la sesión en la BD. Si el registro es exitoso retorna el valor booleano true.
string	<b>ObtEspecialista (string usuario)</b>	Realiza una consulta a la BD en base al campo usuario del paciente que recibe como parámetro para obtener el nombre de especialista.
string	<b>ObtDestino (string usEsp)</b>	Realiza una consulta a la BD en base al nombre del especialista para retornar el email al que se envía la información de la sesión de rehabilitación al especialista cada vez que el paciente realice una rutina.
string	<b>Mensaje (string destino, DatosSesion inf)</b>	Estructura el destino y el mensaje que se enviará para que el especialista conozca la información de la rutina realizada por el paciente.

### 3.2.7 Diagrama de Clases de la Aplicación.

Como se puede apreciar en el diagrama UML de clases de la Figura 71, la aplicación WPF se compone de dieciocho(18) clases. En el apartado anterior se han explicado únicamente ocho y las diez restantes corresponden a las ventanas WPF que conforman la interfaz gráfica de usuario, en las cuales se programa las acciones de algunos controles como botones o el evento load de la ventana y se realiza la llamada a las funciones de las clases creadas.

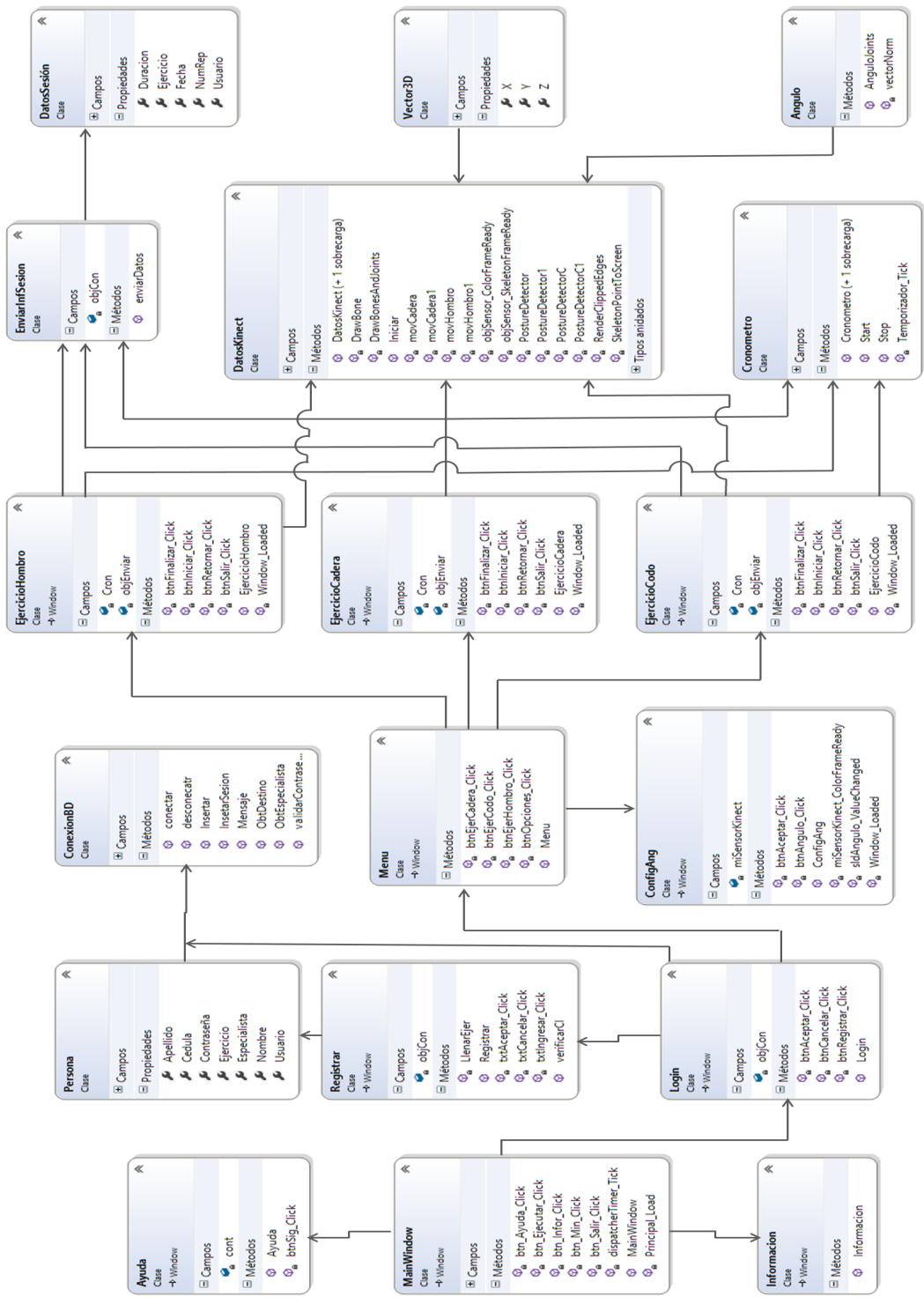


Figura 71: Diagrama UML de Clases .

### 3.2.8 Implementación de la Aplicación Web.

La aplicación en la que el paciente realiza las rutinas de ejercicios permiten almacenar para cada paciente, identificado por el usuario ingresado al iniciar la sesión, los datos relevantes correspondientes a dicha sesión como son el nombre del ejercicio, número de repeticiones que logró el paciente, duración y fecha en la que se realizó el ejercicio. La información es recuperada mediante una aplicación web a la que tiene acceso el especialista que controla el progreso del paciente.

Para el desarrollo de la aplicación Web se ha utilizado el lenguaje PHP, el cual es muy aplicado para la implementación de contenido web dinámico.

#### Interfaz de Usuario de la Aplicación Web.

La interfaz que puede visualizar el usuario se compone de dos ventanas. La primera ventana muestra un entorno con los campos necesarios para poder iniciar sesión y acceder a la información. En caso de que no conste en la base de datos, la ventana permite ingresar los datos necesarios para completar el registro del nuevo usuario. En base a las consideraciones en la etapa de diseño, se implementa la interfaz de la ventana de inicio de la aplicación web, en la que los componentes principales son los botones (Aceptar y Registro) y los campos de texto para ingresar el usuario y contraseña. El archivo creado para este fin se denomina index.php y el aspecto gráfico de la implementación se presenta en la Figura 72.



Figura 72: Interfaz de Usuario de la Aplicación Web (Ventana index.php) .

En la segunda ventana se tienen la presentación de la información del paciente. El componente principal es un combobox, el cual contiene una lista de pacientes recuperada desde la base de datos y una tabla que presenta de forma ordenada la información del paciente seleccionado.

La ventana se implementó mediante un formulario denominado informaci3n.php y cuyo diseño e implementación se tiene en la Figura 73.

Cerrar Sesión  
Bienvenid@: *Especialista*

Seleccione el Nombre del Paciente:

El nombre seleccionado es: *Usuario*

**LAS SESIONES REALIZADAS SE PRESENTAN A CONTINUACIÓN:**

INFORMACIÓN DE LAS SESIONES					
ID-SESION	USUARIO	EJERCICIO	NUM.REPETICIONES	DURACION	FECHA
1	Usuario	RehabilitarHombroDerecho	0	00:05	07/11/2015
18	Usuario	RehabilitarHombroDerecho	0	00:04	08/11/2015
19	Usuario	RehabilitarCadera	7	02:43	12/11/2015
20	Usuario	RehabilitarHombroDerecho	2	01:34	21/11/2015
21	Usuario	RehabilitarHombroDerecho	2	01:34	21/11/2015
22	Usuario	RehabilitarCadera	4	01:47	21/11/2015
23	Usuario	RehabilitarHombroDerecho	6	01:03	30/11/2015
34	Usuario	RehabilitarHombroDerecho	4	00:25	02/12/2015
35	Usuario	RehabilitarCadera	3	00:00	02/12/2015

Figura 73: Interfaz de Usuario de la Aplicación Web(Ventana información.php) .

### Conexión a la Base de Datos.

La conexión a la base de datos desde la aplicación web es muy importante, ya que es a quien se realizarán las consultas en base a los datos ingresado o seleccionados en la interfaz de usuario, el archivo php que se ha implementado para establecer la conexión se denomina connect\_db.php. Los métodos que intervienen en la conexión son lo que se detallan en la Tabla 24.



Tabla 24:

Métodos Utilizados para la Conexión de la BD.

<b>Método</b>	<b>Descripción.</b>
mysql_connect	Establece la conexión a la base de datos, definiendo como parámetros la ubicación de la base de datos, el usuario y contraseña.
mysql_select_db	Define el nombre de la base de datos y la conexión creada.

### **Validación para el Acceso a la Aplicación Web.**

Para verificar la contraseña ingresada al iniciar sesión se utiliza tres formularios php, uno de ellos es Validar.php, en el que se implementa la verificación del correo y contraseña ingresados por el usuario mediante consultas a la base de datos. Si la información es correcta ingresa a la siguiente ventana que es la que proporciona el archivo información.php , mientras que cuando la información es incorrecta el usuario es redireccionado al formulario index.php.

Los otros dos archivos, que son Sesion.php y salir.php, permiten el inicio de sesión y la destrucción de la sesión.

### **Alojamiento Web de la Aplicación.**

PHP es un lenguaje de programación que se ejecuta en el lado del servidor, por lo que en el desarrollo de la aplicación web se utilizó el software XAMPP para convertir a un computador en un servidor web local y de esa manera poder visualizar y manipular las opciones creadas en la aplicación web.

## CAPÍTULO 4

# PRUEBAS DE FUNCIONAMIENTO DEL PROTOTIPO

Para comprobar que la aplicación funciona de forma correcta se han realizado pruebas relacionadas con la funcionalidad que ofrece la aplicación WPF, la distancia a la cual el sensor logra reconocer adecuadamente los movimientos, la ejecución de la rutina de ejercicios por parte del paciente y por último el funcionamiento de la aplicación web.

### **4.1 Pruebas de Funcionamiento del Prototipo.**

En esta prueba se verifica que todas las ventanas que conforman la interfaz del prototipo funcionen de la manera correcta y cumplan con el esquema de diseño con el que fueron implementadas.

#### **4.1.1 Rango de Reconocimiento del Sensor *Kinect*.**

El reconocimiento de los objetos por parte del sensor *Kinect* es de aproximadamente 80 cm como mínimo y de cuatro metros como máximo, pero hay que considerar que pese a que el sensor detecte a un usuario a corta o larga distancia el rango de reconocimiento de las articulaciones varía y no puede aportar los datos correctos de las

coordenadas. Es por ello que se debe probar y establece una distancia adecuada a la cual el usuario se pueda ubicar para que el sensor aporte la información para ejecutar la sesión de fisioterapia.

Para establecer una distancia óptima se hace uso de las cámaras de profundidad del sensor *Kinect*, que al contar con un proyector de luces infrarrojas y una cámara que captura los puntos que generan esas luces permiten obtener la distancia de los objetos que se están frente al sensor. El SDK de *Kinect* mediante funciones prediseñadas es el encargado de capturar los puntos del proyector infrarrojo y el tratamiento de los datos se realiza tomando en cuenta que al momento en el que se inicia la captura de datos de profundidad se tiene gran cantidad de distancias como información proveniente del sensor, las mismas que deben ser almacenados en un arreglo de tipo short que tiene un tamaño de 16 bits. Del tamaño total del arreglo, 13 bits corresponden a información de profundidad y los tres restantes se utilizan para la detección de esqueletos(Figura 74).



Figura 74: Estructura de la Información de Profundidad .

Para desarrollar la programación que permite trabajar con los datos de profundidad se debe seguir el mismo proceso que con la cámara de color y contempla los siguientes pasos:

- Habilitar el flujo de información de profundidad.
- Capturar los datos con el Event Handler.
- Manejo de las distancias de profundidad.

```

miSensorKinect.DepthStream.Enable();
miSensorKinect.Start();
miSensorKinect.DepthFrameReady += miSensorKinect_DepthFrameReady;

```

Figura 75: Sentencias para Habilitar la Información de Profundidad .

El código C #WPF implementado para visualizar, mediante una imagen con colores, si el usuario se encuentra muy alejado o demasiado cerca del sensor tiene la lógica de programación que se presenta en la Figura 76.

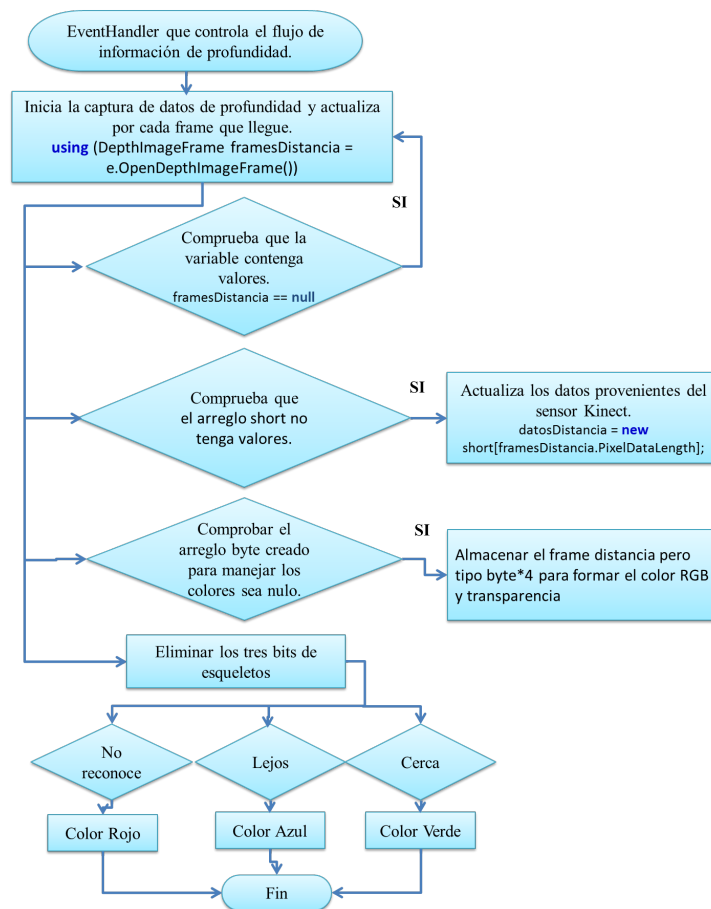


Figura 76: Estructura de la Información de Profundidad .

Al ejecutar la aplicación realizada se puede observar que se presenta una imagen en la que se distingue por colores la escena que se encuentra frente al sensor. Los colores son rojo para los objetos que no son reconocidos, verde para los objetos que

se encuentran muy lejos, azul para los objetos que están muy alejados y un nivel de transparencia para las figuras que se encuentren dentro del rango adecuado de captura.

Las pruebas fueron realizadas con el sensor colocado a una altura de 118 cm con relación al piso y con un ángulo de elevación de  $0^\circ$ , al variar la distancia del usuario con respecto al sensor se obtuvo los resultados gráficos que se presentan en la Figura 77 y que se detalla en la Tabla 25.

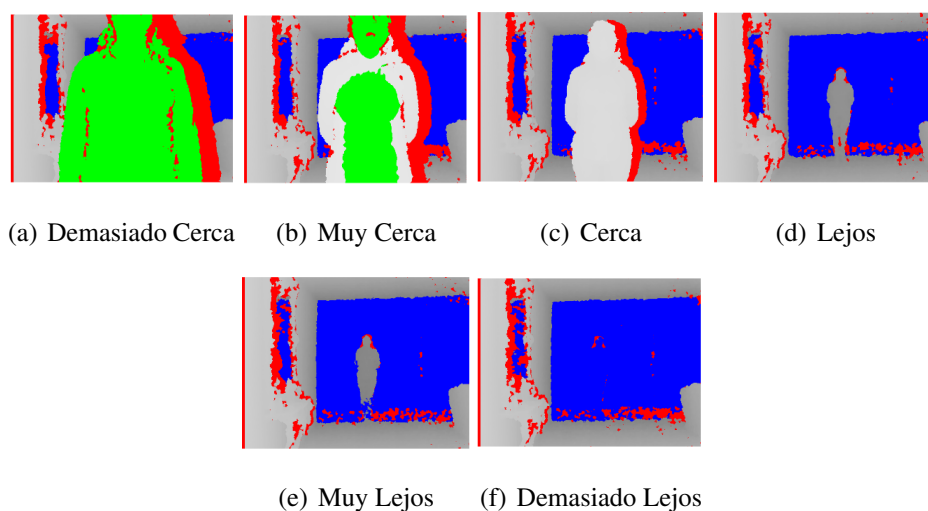


Figura 77: Distancias del Usuario con Respecto al Sensor.

Tabla 25:

Rango de Distancias del Sensor.

Distancia (cm)	Descripción	Color
49	Usuario demasiados cerca del sensor.	Verde
67	Usuario muy cerca del sensor	Verde/Transparencia
105	Usuario cerca del sensor.	Transparencia
334	Usuario lejos del sensor.	Transparencia.
362	Usuario muy alejado del sensor.	Azul/Transparencia
334	Usuario demasiado alejado del sensor.	Azul.

De los resultados obtenidos se determina que a partir de los 105 cm de distancia del sensor se empieza a tener información que puede ser utilizada en aplicaciones en las que se necesite las coordenadas de las articulaciones, esto se identifica por el nivel de transparencia presente en la imagen (c) de la Figura 77, en la que el color verde que representa los puntos más cercanos al sensor desaparece. Por otro lado, la distancia máxima a la cual puede el sensor enviar datos útiles es 334 cm, para distancias mayores se empieza ya a tener datos imprecisos como refleja las imágenes (e) y (f) de la Figura 77, en las que se hace presente el color azul que representa los puntos más alejados del sensor. Por lo que, el usuario debe moverse y ubicarse en el rango de 105 cm a 334 cm de sensor para que pueda obtenerse datos correctos de las coordenadas sin mucha variación.

#### **4.1.2 Funcionamiento de la Aplicación WPF.**

Para comprobar el funcionamiento de la aplicación WPF se realizaron pruebas de cada una de las interfaces desarrolladas, verificando que cada una cumpla con el objetivo con el que fueron creadas. A continuación se presentan los resultados de la verificación de la funcionalidad de las interfaces más importantes de prototipo.

#### **4.1.3 Ventana Principal.**

Al ejecutar la aplicación se comprueba que todos los componentes de la ventana están operativos, ya que la sincronización de fecha y hora de aplicación con el sistema se realiza correctamente. Además, la funcionalidad de los botones también es correcta. La prueba de funcionamiento se resume en la Tabla 26.

Tabla 26:

Verificación de Funcionamiento de la Ventana Principal.

Funcionamiento	Verificación	Observación
Fecha/Hora	✓	Ninguna
Botones Ejecutar, Información y Ayuda	✓	Ninguna

#### 4.1.4 Iniciar Sesión y Registrar.

Para realizar la ejecución de la rutina se debe iniciar sesión en la que se ingresa el usuario y contraseña, si la información es correcta ingresa al menú de ejercicios. Si el usuario no está registrado debe realizar el registro correspondiente. El funcionamiento de estas interfaces es correcto y muestra los mensajes adecuados para informar al usuario lo que ocurre, como puede verse en Figura 78 y en la Tabla 28.

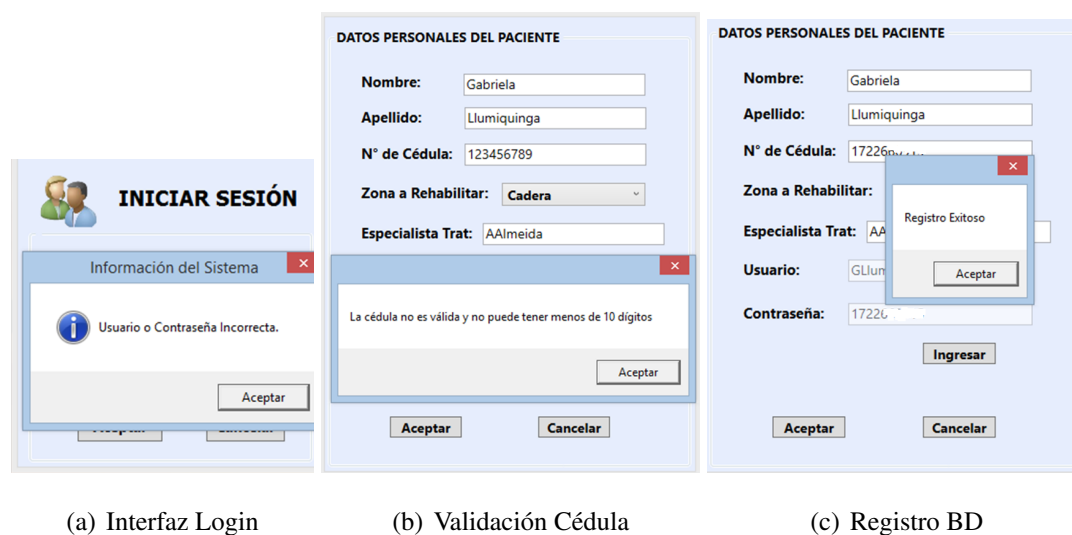


Figura 78: Mensajes Informativos de las Interfaces Login y Registrar.

Tabla 27:

Verificación de Funcionamiento de las Interfaces Iniciar Sesión y Registrar.

<b>Funcionamiento</b>	<b>Verificación</b>	<b>Observación</b>
Acceso al Menú	✓	Ninguna
Aviso Usuario/Contraseña incorrecto	✓	Ninguna
Validación de CI	✓	Ninguna
Registro en la BD	✓	Ninguna

#### 4.1.5 Ejecución de la Rutina de Ejercicios.

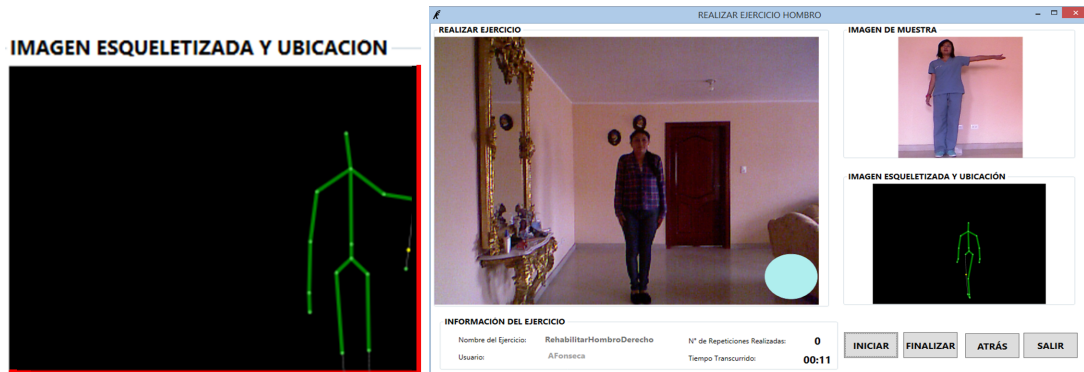
Una vez que el usuario ha ingresado de manera exitosa al sistema se presenta el menú con los ejercicios disponibles. En el entorno gráfico del menú también se presenta un campo de texto con el usuario que ha ingresado a la aplicación, componente que en las pruebas realizadas funcionó exitosamente.

Al seleccionar un ejercicio e ingresar en el entorno para realizar la sesión el usuario inicia el cronometro y observa en la imagen de muestra con la que cuenta el prototipo, los movimientos que debe realizar. El número de repeticiones depende del número de movimientos con los que cuenta el ejercicio, al realizar el último de los movimientos se incrementan las repeticiones y se regresa a la primera postura; para conocer que se realiza correctamente los movimientos el usuario puede observar en la pantalla un indicador que cambia de color cuando la postura es correcta. Al finalizar la rutina la información de las repeticiones realizadas y el tiempo que duró la sesión es almacenada en la base de datos y enviada al correo electrónico del terapeuta.

Para comprobar el funcionamiento, se solicitó a un usuario que seleccione un ejercicio, inicie la rutina, se ubique dentro del rango de distancias que se obtuvo en la sección 4.1 (buscando que el sensor identifique todo el cuerpo humano) y realice la rutina. El usuario seleccionó el ejercicio para rehabilitar el hombro, el cual consta de



dos movimientos para completar una repetición, los resultados gráficos obtenidos de la prueba se presentan en la Figura 79.

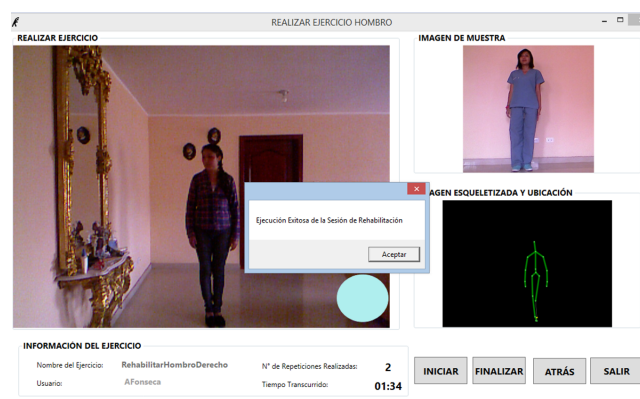


(a) Error en la Ubicación

(b) Movimiento 1



(c) Movimiento 2



(d) Finalizar

Figura 79: Ejecución de la Rutina de Ejercicio.

En la imagen (a) de la Figura 79 se puede observar que el sensor no puede identificar completamente las partes de cuerpo y se presentan áreas de color rojo sobre la imagen esqueletizada que orientan al paciente a dónde debe moverse para lograr una ubicación adecuada. Como se puede observar en los literales (b) y (c) de la Figura 79, los movimientos que debe realizar el paciente son mostrados con imágenes que van cambiando cuando la postura realizada es correcta y cuando el indicador cambia de color para informar al paciente que ha alcanzado un nivel aceptable de acierto en la ejecución. Al finalizar la rutina se presenta un mensaje que informa que la ejecución fue correcta como se puede observar en la imagen (d) de la Figura 79, además el cuadro de diálogo es un indicador de que la información de la rutina ha sido registrada en la base de datos y que el correo electrónico fue enviado. Para comprobar que el mensaje llegó a la dirección de correo del terapeuta, almacenada en la base de datos, y para verificar la estructura del mismo se ingresa a la bandeja de entrada, buscando un mensaje proveniente de un correo denominado *Prototipo Auxiliar para Fisioterapia* que es el que se creó para este fin y el asunto *Ejecución de Rutina*. En la Figura 80 y Figura 81 se puede observar la forma en la que se presenta en la bandeja de entrada y la estructura que tiene el mensaje.

<input type="checkbox"/>	● Prototipo Auxiliar p	Ejecución de Sesión	El usuario AFonseca realizó el ejercicio RehabilitarCadera. t
<input type="checkbox"/>	● Prototipo Auxiliar p	Ejecución de Sesión	El usuario AFonseca realizó el ejercicio RehabilitarHombro/
<input type="checkbox"/>	● Prototipo Auxiliar p	Ejecución de Sesión	El usuario AFonseca realizó el ejercicio RehabilitarHombro/

Figura 80: Mensaje Enviado-Bandeja de Entrada .

- **Prototipo Auxiliar para Fisioterapia**  
 El usuario AFonseca realizó el ejercicio RehabilitarHombroDerecho. Completando la siguiente información:  
 Número de Repeticiones:2  
 Duración de la Sesión: 01:34  
 Fecha de Ejecución de la Sesión: 21/11/2015

Figura 81: Estructura del Mensaje Enviado .

Después de realizar la prueba con el ejercicio de hombro del cual se realizó la

explicación y de probar muchas repeticiones con los ejercicios de cadera y codo se puede decir, desde el punto de vista de funcionamiento, que el prototipo cumple con las expectativas que se tuvieron desde el inicio del proyecto logrando los resultados que se presentan en la tabla de verificación (Tabla 28).

Tabla 28:

Verificación de Funcionamiento de la Rutina de Ejercicios.

Funcionamiento	Verificación	Observación
Funcionamiento adecuado del cronómetro.	✓	Ninguna
Reconocimiento de Posturas.	✓	Ninguna
Movimientos de muestra.	✓	Ninguna
Número de Repeticiones.	✓	Ninguna
Indicador de ejecución correcta.	✓	Ninguna
Imagen esqueletizada	✓	La imagen tarda en aparecer cuando el usuario se encuentra cerca o demasiado cerca del sensor.
Verificación del área de captura.	✓	Ninguna
Registro en la BD de la información de la rutina	✓	Ninguna
Envío del mail.	✓	Puede tardar unos segundos hasta que se envié el mensaje.
Funcionamiento de todos los componentes de la interfaz	✓	Ninguna

#### 4.1.6 Funcionamiento de la Aplicación Web.

Con el fin de probar el funcionamiento de la aplicación web se realizó una rutina con cada ejercicio y con usuarios diferentes. Los resultados de la sesión se enviaron al servidor con el fin de constatar que datos se perdieron y cuales llegaron con éxito. Las pruebas realizadas consistieron en:

- Prueba 1.- se realizó el ejercicio de hombro y se envió los datos de la sesión.
- Prueba 2.- se realizó el ejercicio de cadera y se envió los datos de la sesión
- Prueba 3.- se realizó el ejercicio de codo y se envió los datos de la sesión
- Prueba 4.- verificar que en la lista desplegable se muestre todos los nombres de usuario de los pacientes asignados al terapeuta que ingresa a la aplicación web. Se tuvo la ayuda de tres pacientes que utilizaron el prototipo y que enviaron la información de la sesión.
- Prueba 5.- verificar que la información de la tabla que se muestra en la aplicación web sea ordenada y completa.

Los resultados de las pruebas de funcionamiento de la aplicación web se presentan en la Tabla 29.

Tabla 29:

Pruebas de Funcionamiento de la Aplicación Web.

<b>Parámetro</b>	<b>Prueba 1</b>	<b>Prueba 2</b>	<b>Prueba 3</b>	<b>Prueba 4</b>	<b>Prueba 5</b>
Datos de Sesión Enviados	5	5	5	3	6
Datos de Sesión Recibidos	5	5	5	3	6
Porcentaje de Éxito	100%	100%	100%	100%	100%

#### 4.1.7 Evaluación del Sistema.

Por ser un prototipo que sirve de auxiliar de fisioterapia es importante verificar su funcionamiento y efectividad. Una de las pruebas consiste en la evaluación de la interfaz de usuario, la misma que se lleva a cabo en base a una encuesta que se realiza considerando las heurística de Nielsen: Visibilidad del Estado del Sistema (1), Utilización de Lenguaje de Usuario(2), Control y Libertad para el Usuario(3), Consistencia y Estándares(4), Prevención de Errores(5), Minimización de la Carga Cognitiva(6), Flexibilidad y Eficiencia de Uso(7), Ayuda y Documentación(8), Diseño Minimalista (9) y Ayuda a los Usuarios ante los Errores(10). Los resultados que se presentan en la Figura 83 son obtenidos mediante la aplicación de la encuesta a un total de cinco personas. El modelo de encuesta se presenta en la Figura 82.

UNIVERSIDAD DE LAS FUERZAS ARMADAS-ESPE			
EVALUACIÓN DE LA INTERFAZ NATURAL DE USUARIO DEL PROTOTIPO PARA FISIOTERAPIA CON KINECT			
<i>Indicación.- Marque con una X el casillero de la respuesta que considere adecuada.</i>			
N°	DESCRIPCIÓN DE LA PREGUNTA	SI	NO
1	¿El sistema informa a los usuarios su estado?		
2	¿El sistema utiliza un lenguaje entendible para los usuarios?		
3	¿El sistema cuenta con las opciones de atrás y regresar para proveer al usuario una salida fácil?		
4	¿El usuario entiende el significado de las situaciones o acciones del sistema?		
5	¿El sistema cuenta con prevención de errores y pregunta al usuario si está seguro de realizar una determinada acción?		
6	¿El sistema tiene instrucciones visibles o están al alcance cuando el usuario las requiera?		
7	¿El sistema tiene acciones para acelerar el uso de éste?		
8	¿La interfaz tiene diálogos estéticos y diseños minimalistas?		
9	¿Los mensajes de erros se explican en lenguaje claro y entendible para el usuario?		
10	¿El sistema ofrece ayuda y documentación?		

GRACIAS POR SU COLABORACIÓN

Figura 82: Modelo de Encuesta.

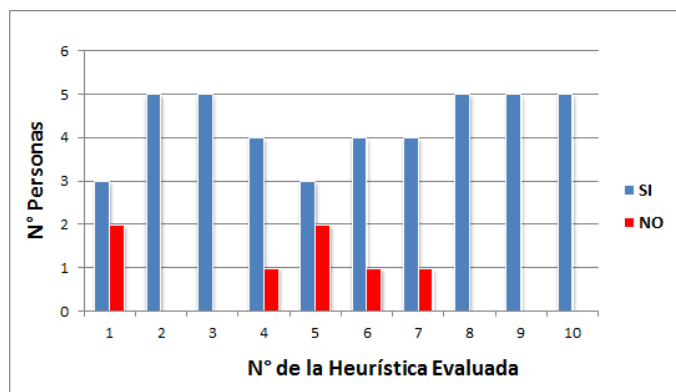


Figura 83: Resultados de la Encuesta Aplicada en Base a los Parámetros de Nielsen.

Los resultados reflejan la conformidad del usuario. El punto 1 resulta algo restrictivo debido a que existe cierto retardo al realizar el trackeo de las articulaciones por parte del sensor. El comportamiento del aspecto 5 se debe a que los usuarios olvidaban conectar el sensor kinect al PC. Mientras que los aspectos 6 y 7 no tuvieron una ponderación excelente debido a que dos de los usuarios encuestados no tenían experiencia en computación.

## 4.2 Prueba de Rehabilitación Piloto.

Con esta prueba se tiene un panorama con el que se puede evaluar el progreso de la sesión de fisioterapia realizada con ayuda del prototipo.

### 4.2.1 Escenario de Prueba

Para la evaluación del prototipo con un paciente en proceso de rehabilitación se ha establecido un escenario de pruebas que se muestra en la Figura 84.

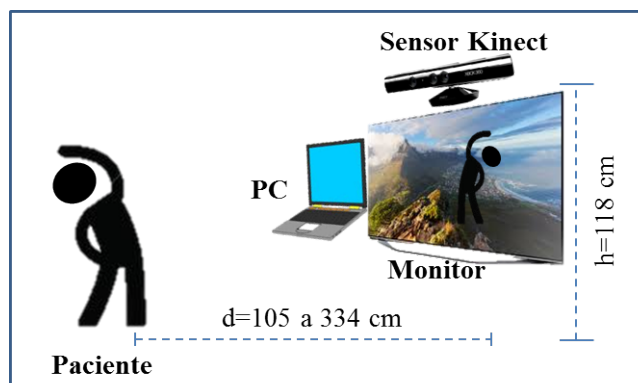


Figura 84: Esquema del Escenario de Prueba .

Las sesiones de prueba se realizó con un paciente que presenta las características que se detallan en la Tabla 30.

Tabla 30:

Datos del Paciente.

<b>Edad:</b>	51 años
<b>Género</b>	Masculino
<b>Diagnóstico</b>	Artritis Reumatoidea
<b>Zona a Rehabilitar</b>	Hombro Derecho
<b>Objetivo de Ejercicios Recomendados</b>	Fortalecimiento Muscular
<b>N° de Repeticiones Recomendadas</b>	10 repeticiones por día.
<b>N° de Sesiones</b>	Mínimo 10 Máximo 30 sesiones.

#### 4.2.2 Consideraciones de la Prueba

De acuerdo al escenarios planteado en la Figura 84, el usuario debe ubicarse frente al sensor a una distancia de 105 a 334 cm para lograr una esqueletización completa del cuerpo humano, la altura a la que esta ubicado el *Kinect* es 118 cm desde el piso y el ángulo de elevación es de cero grados. La aplicación se ejecuta desde una PC y los movimientos pueden visualizarse en el monitor de la misma PC o en uno externo

como un televisor.

Como ya se ha descrito en el objetivo del ejercicio en la Tabla 30, el paciente debe realizar la rutina para retardar la pérdida de la función articular y ayudar a mantener los músculos fuertes. Se ha seleccionado el ejercicio de hombro derecho, ya que el usuario presenta molestia y pérdida de movilidad en la articulación del hombro.

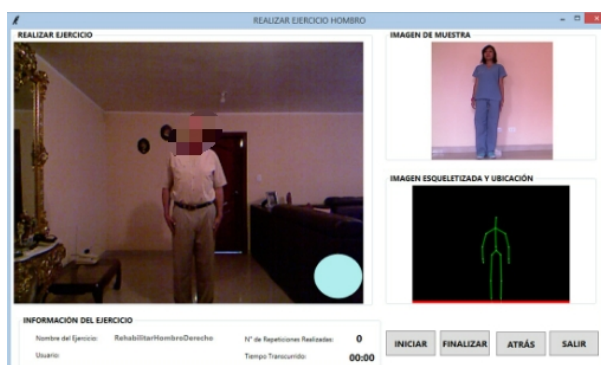
La terapia señalada por un terapeuta consisten en alcanzar diez repeticiones en cada sesión por un período de tiempo de alrededor de un mes, las cuales se realizaran con ayuda del prototipo y en las que se evaluará el número de repeticiones alcanzadas y la duración de la sesión.

### **4.2.3 Pruebas**

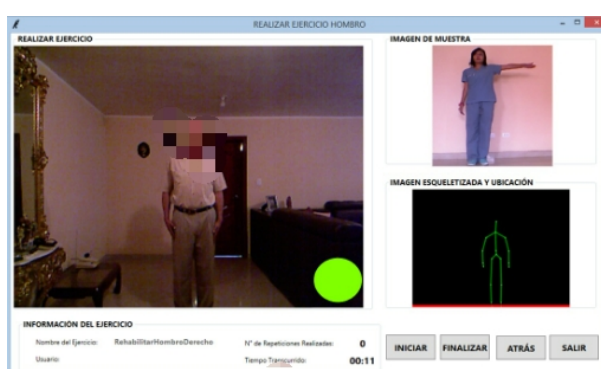
La prueba fue realiza en un ambiente cerrado con unas dimensiones de aproximadamente 2.5 m de ancho y 3.5 m de largo, con una iluminación adecuada para que pueda identificarse la imagen que capta el sensor.

Como se puede apreciar en la Figura 85, el paciente realiza las rutinas para rehabilitar el hombro derecho con ayuda del prototipo auxiliar para fisioterapia, la misma que consta de dos movimientos para completar una repetición. La ejecución correcta de las posturas se puede conocer observando el circulo de la parte inferior derecha, ya que este se pone de color verde cada vez que se alcanza un nivel adecuado y continúa a la siguiente postura.

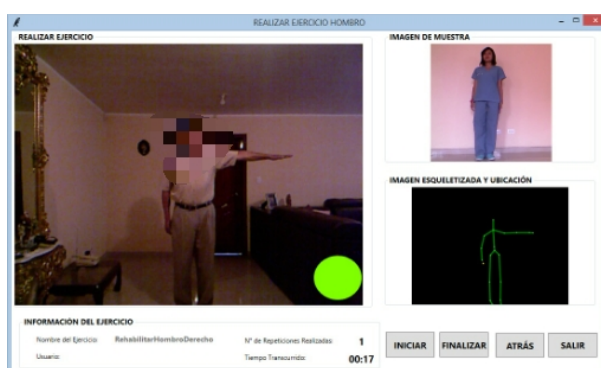




(a) Inicio de la Rutina



(b) Ejecución del Primer Movimiento



(c) Ejecución del Segundo y Último Movimiento

Figura 85: Paciente en la Prueba Piloto.

En la Figura 85 (a) se puede observar que el usuario inicia la ejecución de la rutina y la imagen de muestra señala que el paciente debe realizar un movimiento inicial denominado Neutro. Mientras que en la Figura 85 (b) se puede apreciar que el usuario

ha ejecutado correctamente el movimiento inicial, por lo que el círculo de la parte inferior derecha de la imagen RGB es de color verde, y la imagen de muestra cambia señalando la siguiente postura. Finalmente la Figura 85 (c) muestra que el segundo y último movimiento del ejercicio realizado por el paciente es correcto, por lo que se enciende el bombillo verde, se incrementa el contador de repeticiones y se marca el tiempo transcurrido hasta completar el ciclo que comprende la culminación de los dos movimientos que componen este ejercicio.

#### **4.2.4 Resultados de la Prueba Piloto del Prototipo**

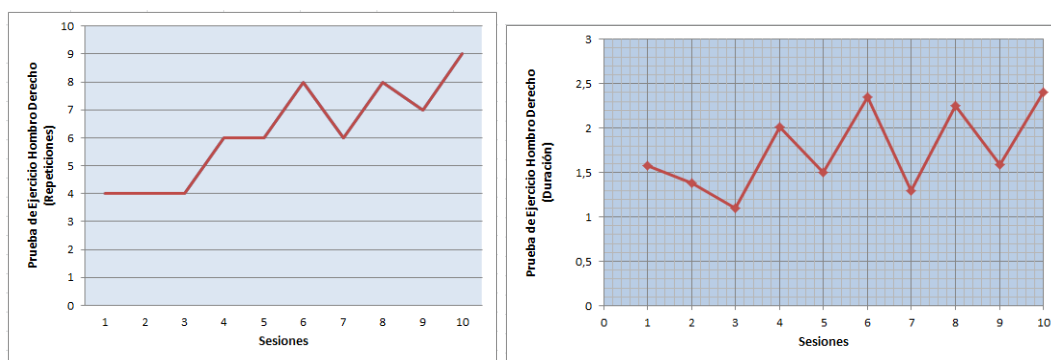
Las valoraciones obtenidas de la ejecución de la rutina con el ejercicio "HombroDerecho" para cada una de las sesiones realizadas se presenta en la Tabla 31. En la que se aprecia que el usuario no ha logrado alcanzar la meta diaria señalada por el especialista, ya que se han presentado ciertos factores como dolor o fatiga que han ocasionado que el paciente abandone la sesión y puesto que la rehabilitación es un proceso que se realiza paulatinamente los resultados se observan a largo plazo. Además, se puede notar que el tiempo se ha incrementado en alguno casos al igual que el número de repeticiones.

Tabla 31:

Datos Obtenidos de la Prueba Piloto.

N° Sesión	N° Repeticiones	Duración (min)
1	4	1,58
2	4	1,38
3	4	1,1
4	6	2,01
5	6	1,5
6	8	2,35
7	6	1,3
8	8	2,25
9	7	1,59
10	9	2,4

Para el análisis de la prueba piloto se ha tomado en cuenta diez sesiones realizadas con ayuda del prototipo, los resultados de dichas sesiones se puede observar en las representaciones gráficas de la Figura 86, basándose en el número de repeticiones y el tiempo que ha tardado el paciente en completar la sesión.



(a) N° Repetición

(b) Duración

Figura 86: Progreso del Paciente en la Prueba Piloto.

La Figura 86 (a) relaciona el número de repeticiones y el avance de las sesiones realizadas, de tal manera que se puede determinar que existe una tendencia ascendente en el progreso de la rehabilitación, ya que el paciente va aumentando el número de repeticiones en cada sesión, logrando en las tres primeras sesiones realizar únicamente cuatro repeticiones debido a las molestias y dolor que presentaba el paciente en la articulación. Cuando se alcanzó la quinta sesión se consiguió seis repeticiones y el paciente se encontraba más familiarizado con el sistema, aunque al finalizar la rutina el paciente se notaba fatigado y con una mínima molestia en la articulación que al cabo de algunos segundos se aliviaba, por lo que lograba ocho repeticiones. En la décima sesión el paciente ejecutaba la rutina sin ninguna alteración, realizando nueve repeticiones.

En cuanto al tiempo de duración, se aprecia en la Figura 86 (b) que varía dependiendo de la cantidad de repeticiones que consigue el paciente, es así que las sesiones en las que se realizaban cuatro repeticiones el tiempo se redujo de 1.58 segundo a 1.1 segundos, cuando se lograban seis repeticiones el tiempo cambió de 2.01 a 1.3 segundos y al ejecutar ocho repeticiones hubo una disminución de 2.35 a 2.25 segundos. Por lo que se puede decir que mientras las sesiones fueron avanzando el tiempo se disminuye al ejecutar el mismo número de repeticiones que en sesiones anteriores, esto se debe a que cuando más se ejercita la articulación va adquiriendo mayor movilidad y el dolor se va disminuyendo, lo que ocasiona que el paciente realice el mismo número de repeticiones en menor tiempo. También se presenta el caso que se tienen en la sesión seis y diez, en el que tiempo de duración es el mismo pero el número de repeticiones se ha incrementado en la décima sesión.

De resultados obtenidos de la prueba piloto, se puede decir que existe un progreso en las condiciones de movilidad y fuerza muscular del paciente, por lo que el prototipo cumple con el objetivo de mejorar la calidad de vida de los usuarios. En este caso, la artritis reumatoidea requiere tratamiento de por vida que consiste en medicamento, cirugía, ejercicio y fisioterapia, por lo que se requiere de sesiones terapéuticas constantemente y en las cuales el prototipo sería un buen aliado.

# CAPÍTULO 5

## CONCLUSIONES Y RECOMENDACIONES

### 5.1 Conclusiones

Uno de los aspectos más importantes para el desarrollo de una aplicación con *Kinect* es la selección de la versión del *Kit de Desarrollo de Software (SDK)*, que mediante una exhaustiva investigación y pruebas realizadas con las últimas versiones (SDK 1.7, SDK 1.8 y SDK 2.0) en el desarrollo del proyecto se llega a la conclusión de que su selección depende de la versión de Visual Studio que se va a utilizar. De ahí que en la ejecución del proyecto se utilizó Visual Studio 2013, en el cual se probó las tres versiones del SDK, resultando completamente funcional el SDK 1.7 que se caracteriza por ser una de las versiones con mayor estabilidad.

Al contar con la versión adecuada del SDK, se realizó un estudio del mismo para determinar las opciones que brinda y a las cuales se puede acceder en el desarrollo de aplicaciones con *Kinect*, lográndose habilitar las funciones necesarias para inicializar el sensor y obtener la información de profundidad, imágenes de color e información de esqueletos necesarias para el desarrollo del proyecto.

La validación de los movimientos que conforman la rutina de ejercicios de rehabil-

itación se desarrolló después de realizar un estudio y breve revisión de los conceptos de fisioterapia que involucran ejercicios terapéuticos activos libres con movimientos de aducción y abducción, que permitieron establecer los parámetros necesarios como ángulos y posición de las articulaciones que intervienen en el movimiento para poder realizar una identificación de posturas con el sensor *Kinect*.

Durante el desarrollo de la aplicación debido a que el sensor proporciona información de movimientos en tiempo real, se tuvo dificultad para controlar el contador del número de repeticiones, ya que no se tenía una identificación de posturas adecuada y por lo cual el contador se incrementaba aceleradamente. Para solucionar el inconveniente se planteó la posibilidad de una programación paralela, que consiste en utilizar dos procesadores para cubrir una tarea compleja que requiere que muchas instrucciones se ejecuten simultáneamente, esta opción fue descartada debido a que requiere un alto costo computacional y un alto consumo de energía. La alternativa más viable y la que fue seleccionada consiste en un identificador de posturas en el que se define el movimiento. Para incrementar el número de repeticiones el prototipo debe identificar como válido el movimiento, para lo cual se establece un detector de posturas que se basa en la evaluación de un dato de tipo enumerador para determinar si es ejecutada de forma correcta o caso contrario establecer como una postura nula, lográndose un funcionamiento óptimo del prototipo que permite una adecuada ejecución de la rutina de rehabilitación.

Se ha diseñado y desarrollado los bloques funcionales para el prototipo de fisioterapia con el sensor *Kinect*, el cual sirve de ayuda para las sesiones de rehabilitación que debe realizar un paciente en tratamiento terapéutico para mejorar la movilidad y la fuerza muscular en las articulaciones por medio del ejercicio terapéutico. EL prototipo dispone de acceso a correo electrónico para remitir al terapeuta los resultados de la sesión en los que se detalla las valoraciones (fecha, duración y el número de repeticiones) logradas por el paciente al terminar la rutina de rehabilitación

Se ha dotado la posibilidad de manejar por voz la interfaz del prototipo gracias al arreglo de micrófonos del sensor kinect y mediante la implementación de algoritmos de reconocimiento de voz, lográndose buenos resultados. Por lo que se deja la posibilidad de desarrollar aplicaciones que permitan ejecutar terapias de lenguaje.

Una vez realizadas las pruebas con un paciente que sufre de artritis reumatoidea, quien se adaptó fácilmente a la nueva forma de rehabilitación y se familiarizó con la interfaz del prototipo por el lapso de diez sesiones logrado mejoras en la movilidad, se concluye que los objetivos propuestos en el presente proyecto se han cumplido satisfactoriamente, puesto que este prototipo puede servir como auxiliar en el proceso de rehabilitación de pacientes con problemas en la movilidad de las extremidades superiores o inferiores.

Al momento la aplicación cuenta con tres tipos de movimientos propios de los ejercicios terapéuticos adecuados para tratar tres articulaciones específicas: hombro, cadera y codo, razón por la que se podrían incluir en versiones futuras del prototipo nuevos movimientos para tratar otro tipo de articulaciones como la articulación de la rodilla, el pie o la muñeca. Otra mejora que podría darse al prototipo es la grabación de rutinas por parte del terapeuta, de forma que el paciente pueda tener una lista de ejercicios personalizados y acordes a sus necesidades.

A la hora de captar los movimientos se evitan posturas que pueden obstaculizar la visión del sensor de ciertas partes del cuerpo del usuario. Esto complica la implementación de movimientos en los que intervienen desplazamientos en varias direcciones. Considerando que el SDK oficial de *Kinect* permite trabajar con más de un sensor conectados a un mismo ordenador se propone como mejora del prototipo utilizar dos o más sensores *Kinect* para minimizar este problema y lograr una mejor captación y reconocimiento de posturas complejas. Con esta mejora se podría tratar a personas que han tenido accidentes cerebrovasculares o que tienen defectos congénitos como Síndrome de Down, las cuales necesitan mejorar la postura, el equilibrio y en muchos

casos requieren ser reeducados para mejorar la forma de caminar y movilizarse.

## 5.2 Recomendaciones

Es importante que al momento de desarrollar aplicaciones con el sensor *Kinect* para Xbox se cuente también con el adaptador que permite regular el voltaje necesario para el funcionamiento del sensor. En caso de que se utilice un sensor *Kinect* para *Windows* éste adaptador no es necesario.

Se recomienda verificar que el sensor se encuentre conectado correctamente al computador, ésto se puede conocer observando que el led de color verde ubicado cerca a las cámaras del sensor se encuentre parpadeando, que es un proceso normal y no debe ser causa de temor.

Antes de desarrollar con *Kinect* se debe asegurar que el SDK seleccionado tenga disponibles todas las funciones para la versión de Visual Studio en el que se va a desarrollar la aplicación, realizando pruebas con las diferentes versiones del kit de desarrollo hasta encontrar la adecuada.

Para la elaboración del prototipo se ha utilizado el software de desarrollo oficial y el sistema operativo *Windows 8*, pero existen otras alternativas open source como OpenKinect u OpenNI SDK, con las cuales se puede investigar y trabajar con otros sistemas operativos.

Al momento de realizar la rutina el usuario debe situarse a una distancia adecuada o dentro del rango de distancias definido en el capítulo de las pruebas (105 cm - 334 cm), para que el sensor pueda detectar el cuerpo completo y generar una esqueletización correcta con la información apropiada para realizar la validación de los movimientos.



# BIBLIOGRAFÍA

- ALBIOL PÉREZ, S. (2014). *Rehabilitación virtual motora: una evaluación al tratamiento de pacientes con daño cerebral adquirido*. (Unpublished doctoral dissertation).
- AMarcos(CosasdeSalud). (n.d.). *Kinect en la rehabilitación de la esclerosis múltiple*. Recuperado de: <http://www.cosasdesalud.es/kinect-en-la-rehabilitacion-de-la-esclerosis-multiple/>.
- Arbona, C. B., García-Palacios, A., & Baños, R. M. (2007). Realidad virtual y tratamientos psicológicos. *Cuadernos de medicina psicosomática y psiquiatría de enlace*(82), 17–31.
- Beltrán, F. F. G., Cea, & López. (n.d.). *Mocap principales tecnologías*. Recuperado de: <http://sabia.tic.udc.es/gc/Contenidos20adicionales/trabajos/Peliculas/Mocap/tecnol.html>.
- BRAVO, J. L., TORDESILLAS, M. J., PADRÓN, M. A., JEREZ, N. A., GONZÁLEZ, V., & BLANCO, A. (n.d.). Plataforma accesible en el marco de la rehabilitación físico-cognitiva.
- Cobo, A. (2005). *Php y mysql: Tecnología para el desarrollo de aplicaciones web*. Ediciones Díaz de Santos.
- CONADIS. (2012). *Agenda nacional para la igualdad en discapacidades 2013-2017*.
- Fernández Sánchez, E. (2012). Control de software educativo mediante kinect de microsoft.
- Gallego, T. (2007). Bases teóricas y fundamentos de la fisioterapia. *Ed Mádica Panamericana*.

- Hernández Toala, L. A., & Herrera Rodríguez, J. D. (2013). *Análisis y estudio de los códigos fuente sdk (kit de desarrollo de software) e implementación de una aplicación demostrativa que registre la captación de movimientos de manos y brazos del cuerpo humano a través de leds indicadores mediante la utilización del sensor kinect del xbox 360*. (Unpublished doctoral dissertation).
- Ilvay Taday, R. B. (2014). Sistema de educación para niños de 3 a 5 años, mediante un robot controlado por el sensor kinect.
- INEC. (2010). *Mujeres y hombres del ecuador en cifras iii* (Vol. 3). ECUADOREN-CIFRAS.
- InstitutodeBiomecánicadeValencia. (n.d.). *Nedsve/ibv-aplicación para la valoración y rehabilitación del equilibrio*. Recuperado de: <http://www.ibv.org/productos-y-servicios/productos/aplicaciones-biomecnicas/nedsveibv-aplicacion-para-la-valoracion-y-rehabilitacion-del-equilibrio>.
- Katić, D., Ćosić, A., Despotović, T., & Miloradović, B. (n.d.). Quickbot as educational and research platform for multi mobile robotic systems.
- Kitagawa, M., & Windsor, B. (2012). *Mocap for artists: workflow and techniques for motion capture*. CRC Press.
- LUMO. (2012). *Lumo back*. Recuperado de: <http://www.lumobodytech.com/lumo-back/>.
- Microsoft. (n.d.). *Kinect for windows architecture*. Recuperado de: <https://msdn.microsoft.com/en-us/library/jj131023.aspx>.
- Milgram, P., & Colquhoun, H. (1999). A taxonomy of real and virtual world display integration. *Mixed reality: Merging real and virtual worlds*, 5–30.
- Moreno, F., Jordan, O., Esmitt, R. J., Christiam, M., Omaira, R., Jeanlight, R., & Silvio, Á. (2013). Un framework para la rehabilitación física en miembros superiores con realidad virtual. In *Primera conferencia nacional de computacion, informatica y sistemas. universidad central de venezuela*.
- Nike. (2012). *Nike+fuelband*. Recuperado de: <https://secure-nikeplus.nike.com/plus/products/fuelband/>.

- Nintendo. (n.d.). *Accesorios*. Recuperado de: <http://www.nintendo.com>.
- Nuño Simón, J. (2012). Reconocimiento de objetos mediante sensor 3d kinect. Online, F. (n.d.). *Técnicas de fisioterapia*. Recuperado de: <http://fisioterapiaonline.com/tecnicas/tecnicas.html>.
- Playstation. (n.d.). *Accesorios*. Recuperado de: <http://latam.playstation.com>.
- Robledo Zarco, J. (2012). Prototipado de una aplicación interactiva mediante kinect: motor de captura y reconocimiento de posturas y gestos.
- Rodriguez Estevez, A. (2013). Sistema d'entrenament amb kinect.
- Román Barrera, C., Sanabria Daza, L., Tengono Ramírez, C., et al. (2014). Estudio bibliométrico de la producción científica sobre la aplicación del nintendo wii en miembro superior de personas con deficiencias funcionales asociadas a ecv.
- SANCHIS CANO, Á. (2013). *Mediknect: Un sistema de rehabilitación virtual* (Unpublished doctoral dissertation).
- Suehring, S. (2002). *Mysql bible*. John Wiley & Sons, Inc.
- Toribios Blázquez, J. (2012). Modelado 3d de objetos a partir de la kinect.
- Tortora, G., & Derrickson, B. (2009). *Anatomía y fisiología humana*. Panamericana.
- Velarde, M. P., Perugachi, E., Romero, D., Sappa, A. D., & Vintimilla, B. X. (2015). Análisis del movimiento de las extremidades superiores aplicado a la rehabilitación física de una persona usando técnicas de visión artificial. *Revista Tecnológica-ESPOL*, 28(1).
- Villada, F., & Muñoz, J. (2014). Desarrollo de un software de análisis biomecánico a través de datos de captura de movimiento usando el sensor kinect para rehabilitación asistida con video juegos. *Entre Ciencia e Ingeniería*(16).
- VirtualRehab. (n.d.). Recuperado de: <http://www.virtualrehab.info/es/>.
- XBOX. (n.d.). *Accesorios*. Recuperado de: <http://www.xbox.com>.

**UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE****INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES****ACTA DE ENTREGA**

El presente proyecto fue entregado en el Departamento de Eléctrica y Electrónica, y reposa en los archivos desde:

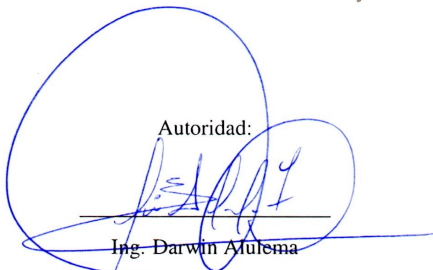
Sangolquí, 22 de enero de 2016.

Elaborado por:



Sonia Alejandra Fonseca Factos

Autoridad:



Ing. Darwin Aulema

DIRECTOR DE LA CARRERA DE INGENIERÍA ELECTRÓNICA Y  
TELECOMUNICACIONES