



**UNIVERSIDAD POLITÉCNICA DE
PUEBLA**

PROGRAMA ACADÉMICO DE
INGENIERÍA EN INFORMÁTICA

**Cinemática Directa para Mover las Articulaciones
de un Actor Digital**

Ignacio Huitzil Velasco

Reporte Técnico PII-01-04-09

COMITÉ EVALUADOR

Dr. Antonio Benítez Ruiz (*Asesor*)
Dr. Jorge de la Calleja Mora (*Sinodal*)
Dra. María Auxilio Medina Nieto (*Sinodal*)

PROFESOR(A) DE PROYECTO DE INVESTIGACIÓN II

Dra. María Auxilio Medina Nieto

Juan C. Bonilla, Puebla

Abril 2009

Índice

Capítulo 1. Planteamiento del Problema de Investigación

1.1. Introducción.....	11
1.2. Objetivo general	13
1.3. Objetivos específicos.....	13
1.4. Justificación	14
1.5. Cronograma de actividades	17
1.6. Alcances y limitaciones	22
1.7. Recursos de hardware y software.....	22

Capítulo 2. Marco Teórico

2.1. Introducción a la cinemática.....	23
2.1.1. Definición de cinemática.....	23
2.1.2. Definición de robótica.	23
2.1.3. Cinemática del robot.....	24
2.1.4. Cinemática directa e inversa.....	25
2.1.5. Transformaciones geométricas para mover cuerpos rígidos en 3D.....	25
2.2. Cinemática	
2.2.1. Uniones revolutivas y prismáticas.....	34
2.2.2. Convención Denavit-Hartenberg.....	35
2.2.3. Articulación esfera-y-cavidad.....	39
2.3. Actor Digital	
2.3.1. Definición de actor digital.	40
2.3.2. El cuerpo humano y sus movimientos.	41
2.3.3. Modelo del actor digital	
2.3.3.1. Modelado del actor digital.....	47
2.3.3.2. El esqueleto del actor digital y sus articulaciones	49
2.4. Trabajo relacionado	51

Capítulo 3. Diseño de la investigación

3.1. Introducción.....	53
3.2. Requerimientos.....	54
3.3. Documento de Requerimientos	55
3.4. Casos de uso y casos de prueba.....	56
3.5. Módulos	63
3.6. Manejo de datos	64

Capítulo 4. Implementación

4.1. Manipulación de archivos .txt	66
4.2. Rutinas de movimiento para las articulaciones	67
4.3. Restablecer interfaz Digital Actor	70
4.4. Componentes de la aplicación Digital Actor.....	71
4.5. Control de articulaciones	75
4.6. Interfaz de la aplicación Digital Actor	75

Capítulo 5. Resultados de “Digital Actor”

5.1. Cabeza	80
5.2. Hombro	80
5.3. Brazo.....	81
5.4. Muñeca	82
5.5. Mano	82
5.6. Tórax.....	83
5.7. Pierna	83
5.8. Pie.....	84

Capítulo 6. Conclusiones

85

Referencias	86
Apéndice	88

Índice de tablas

Tabla 1.1 Actividades para la propuesta de protocolo de investigación.	17
Tabla 1.2 Actividades para migrar el formato del modelo de un actor digital.....	18
Tabla 1.3 Actividades para el estudio de la cinemática y articulaciones.	19
Tabla 1.4 Actividades para codificar rutinas de cada una de las articulaciones del actor digital.....	20
Tabla 1.5 Actividades para implementar la interfaz de controles y análisis de los algoritmos.	21
Tabla 2.1 Rangos de articulación tobillo.....	44
Tabla 2.2 Rangos de articulación codo.....	44
Tabla 2.3 Rangos de Cadera	45
Tabla 2.4 Rangos de articulación rodilla	45
Tabla 2.5 Rangos de articulación de dedo pulgar.....	45
Tabla 2.6 Rangos de articulación dedos de la mano.....	35
Tabla 2.7 Rangos de articulación hombro	46
Tabla 2.8 Rangos de articulación dedos del pie	46
Tabla 3.1 Requerimiento lectura del modelo de un actor digital en formato .txt.....	55
Tabla 3.2 Requerimiento Control y movimiento de 53 articulaciones.....	55
Tabla 3.3 Especificación de actores	62

Tabla 3.4 Módulo Ayuda.....	63
Tabla 3.5 Módulo Control de articulaciones	63
Tabla 4.1 Función de escritura	66
Tabla 4.2 Función de lectura	67
Tabla 4.3 Descripción de funciones que permiten manipular los joints del actor digital.....	68
Tabla 4.4 Definición de las funciones principal y procedimiento	72
Tabla 4.5 Definición de la función CreateWindows y las clases de controles utilizadas.....	73
Tabla 4.6 Definición de la función <i>CreatewindowsEx</i> y las clases de controles utilizadas.	73
Tabla 4.7 Descripción de los controles utilizados en <i>Digital Actor</i>	74
Tabla 4.8 Descripción de la Ventana principal <i>Digital Actor</i>	76
Tabla 4.9 Descripción del menú <i>Control</i> de <i>Digital Actor</i>	76
Tabla 4.10 Descripción del cuadro de diálogo <i>Articulación</i> de <i>Digital Actor</i>	77
Tabla 4.11 Descripción del menú <i>Ayuda</i> de <i>Digital Actor</i>	78
Tabla 5.1 Implementación del movimiento flexión y extensión de cuello.....	80
Tabla 5.2 Resultado de implementar el movimiento rotación hacia adentro o afuera del brazo.	81
Tabla 5.3 Implementación de movimiento pronación y supinación para ambos codos.....	81
Tabla 5.4 Resultado del movimiento flexión y extensión de muñeca izquierda.	82
Tabla 5.5 Resultado del movimiento flexión y extensión de la mano.....	82

Tabla 5.6 Movimiento abducción y aducción de cintura.....	83
Tabla 5.7 Movimiento abducción y aducción de la ingle izquierda.....	83
Tabla 5.8 Resultado de movimiento inversión y eversión activa de ambos tobillos.....	84
Tabla 7.1 Grados de libertad de las articulaciones de <i>Digital Actor</i>	92

Índice de figuras

Figura 1.1 Actor digital que se utilizará en este trabajo	12
Figura 1.2 Articulación Esfera-y-cavidad	15
Figura 2.1 Representación de una cadena cinemática	24
Figura 2.2 Relación entre la cinemática directa e inversa [6]	25
Figura 2.3 Representación de la graficación de un cubo en el Sistema Universal de Referencia	26
Figura 2.4 Sistema coordenado cartesiano	26
Figura 2.5 Matriz neta modelo 1	27
Figura 2.6 Matriz neta modelo 2	28
Figura 2.7 Operaciones de transformación [14]	28
Figura 2.8 Representación de la transformación traslación de un cuerpo rígido [15].....	29
Figura 2.9 Matriz de traslación modelo 1	29
Figura 2.10 Matriz de traslación modelo 2.....	30
Figura 2.11 Matriz de traslación inversa	30
Figura 2.12 Giro de un objeto rígido 3D sobre un eje paralelo al eje x [12].....	31
Figura 2.13 Matrices de rotación para el respectivo eje	31
Figura 2.14 Matriz de transformación escala.	32

Figura 2.15 Escalado de un cuerpo rígido 3D [12].....	32
Figura 2.16 Matriz de transformación homogénea.....	33
Figura 3.1 Articulación revolutiva y prismática.....	35
Figura 3.2 Representación de la convención Denavit-Hartenberg [14]	36
Figura 3.3 Cuatro parámetros de la convención Denavit-Hartenberg [6].	37
Figura 3.4 Unión esfera-y-cavidad [2]	39
Figura 3.5 Implementación de la articulación esfera-cavidad en el humano.	39
Figura 3.6 Matriz para la unión esfera-y-cavidad.....	40
Figura 4.1 Representación de actores digitales.	40
Figura 4.2 Rangos de movimiento del hombro [6] y [8].....	42
Figura 4.3 Rangos de movimiento del codo [6] y [8].....	42
Figura 4.4 Rango de movimiento del tobillo [6] y [8].....	43
Figura 4.5 Rangos de movimiento la rodilla [6] y [8].....	43
Figura 4.6 Rangos de movimiento de la pierna [6] y [8].....	43
Figura 4.7 Rangos de movimiento de la muñeca [6] y [8]	44
Figura 4.8 Diagrama del diseño del modelo [6]	48
Figura 4.9 Modelo del actor digital [4]	48

Figura 4.10 Jerarquía de las articulaciones del esqueleto.....	49
Figura 4.11 Esqueleto y malla de triángulos del actor digital	50
Figura 5.1 Actor digital propuesto por [4] y [6].....	51
Figura 5.2 Actor digital Eugene propuesto por [1].....	52
Figura 6.1 Pantalla de la aplicación Digital Actor.....	54
Figura 6.2 Casos de uso.....	56
Figura 6.3 Módulos de la aplicación Digital Actor	63
Figura 7.1 Esqueleto del actor digital.....	68
Figura 7.2 Función del botón “Restablecer” de la aplicación Digital Actor.	71
Figura 7.3 Ventana principal Digital Actor.	75
Figura 7.4 Ventana principal uso de menú <i>Control</i> Digital Actor.	76
Figura 7.5 Cuadro de dialogo <i>Articulación</i> resultado de la opción Articulaciones.....	77
Figura 7.6 Ventana principal uso de menú <i>Ayuda</i> Digital Actor.	78
Figura 7.7 Cuadro de dialogo <i>Ayuda Digital Actor</i>	79
Figura 7.8 Cuadro de dialogo <i>Acerca de Digital Actor</i> l.....	79
Figura 8.1 Formato de los datos de los triángulos.....	91
Figura 8.2 Formato de los materiales y los joints (caso del joint padre).....	91

Resumen

Este reporte describe la aplicación de técnicas de cinemática directa a cadenas articuladas. Para esto se utiliza una representación gráfica en tres dimensiones de un actor digital, el cual posee un esqueleto que cuenta con 53 articulaciones y 70 grados de libertad. Cada articulación se mueve dentro los rangos permitidos por la anatomía humana. En un trabajo previo se controlaron 14 articulaciones. Una interfaz gráfica en 3D permite la manipulación del movimiento en las articulaciones. El reporte contiene conceptos de cinemática directa, diseño e implementación de las rutinas de animación, resultados de los movimientos asociados a las articulaciones, descripción de las características y funcionalidades de la interfaz, conclusiones y trabajo a futuro.

Capítulo 1. Planteamiento del problema de investigación

1.1. Introducción

Una de las áreas más importantes hoy en día es la robótica menciona [19], la cual permite generar elementos llamados robots. “Un robot puede ser un manipulador multifuncional reprogramable diseñado para mover material, partes, herramientas; también son aparatos especializados que tiene programados movimientos para actuar en una variedad de tareas [13]”, por ejemplo un robot industrial.

Uno de los objetivos de la robótica es crear robots que sean semejantes físicamente y funcionalmente a un humano con cierto grado de autonomía y que puedan ser manipulados mediante programas de computadora. En cuanto al funcionamiento del robot se busca que el movimiento sea lo más cercano al de un humano.

La fabricación de un robot humano tiene diversos inconvenientes como son los factores de tiempo el cual se dedica para realizar los cálculos, la construcción, y la programación, otro factor son los altos costos para su desarrollo. Por estos factores se ha optado por desarrollar simuladores¹ (software) que permitan generar prototipos de robots y mostrar su comportamiento.

Para el desarrollo de este proyecto se cuenta ya con un actor digital, éste es el modelo de una mujer, ver Figura 1, la cual contiene 53 articulaciones, de las cuales 14 ya tienen movimiento. Así, que parte de este proyecto consiste en dar movimiento a las 39 articulaciones restantes, a través de la aplicación de la cinemática directa, también se busca que el movimiento del actor digital coincida con el de un humano, de esta forma se establecerán restricciones cinemáticas para cada articulación del actor digital, de acuerdo a las características físicas del cuerpo humano.

Por lo tanto, se busca generar una simulación de los movimientos del actor digital a través de un ambiente gráfico de computadora en el cual se observará el comportamiento del actor por medio de controles. Este ambiente permitirá a los alumnos que estudien áreas relacionadas a la robótica realizar algunas simulaciones¹.

¹ Es el modelado de situaciones de la vida real por medio de un programa de computadora, para probar el comportamiento.

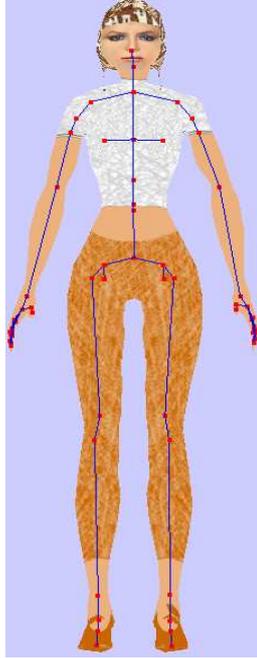


Figura 1.1 Actor digital que se utilizará en este trabajo

1.2. Objetivo general

Aplicar cinemática directa para mover cada una de las articulaciones de un actor digital a través de una interfaz de software, que permita el control de las articulaciones dentro de un ambiente gráfico.

1.3. Objetivos específicos

- Migrar el formato binario (.ms3d) producto de la herramienta MilkshapeModel de un actor digital, a un formato .txt para su mejor compresión y manipulación.
- Analizar la convención y algoritmo de Denavit-Hartenberg [14], y también las uniones revolutiva² y esfera-y-cavidad³ para aplicarlas a las articulaciones del cuerpo humano a través del uso de los *frames*⁴ asociados a cada elemento de la cadena articulada.
- Codificar las rutinas de cada una de las articulaciones del actor digital para poder establecer sus movimientos.
- Implementar una interfaz de software de controles que permitan mover cada una de las articulaciones del actor digital.
- Analizar los algoritmos para lograr que el actor digital camine o se desplace dentro de un ambiente.

² Es una unión que permite la rotación relativa entre dos elementos (eslabones) y tiene un solo grado de libertad.

³ Son uniones esfera- y- cavidad de movimiento rotatorio (un movimiento axial un grado de libertad) y (un movimiento esférico posee dos grado de libertad).

⁴ Es el despliegue gráfico para producir una serie de imágenes

1.4. Justificación

Dentro de la computación, las áreas de graficación y algoritmos aplicados a la robótica y a la simulación han impulsado el desarrollo de una nueva área de aplicación conocida como “actores digitales”[4]. Esta área permite, desde modelar al actor con la forma y comportamientos que el desarrollador desea así como de dotarlo de algunas capacidades de movimiento. Por lo tanto, un actor digital es un ente programado que existe y se mueve dentro de un mundo gráfico virtual [1]. En la actualidad existen diversos actores digitales aplicados en videojuegos, páginas web, en la animación de películas, entre otros.

La importancia de contar con un actor digital con un modelo humano, permite inicialmente estudiar y modelar diferentes técnicas tanto para el diseño del actor como para la representación de cada una de las articulaciones del cuerpo humano a través diferentes clases de uniones (revolutivas y de tipo esfera-y-cavidad)[13]. Por otro lado, es necesario estudiar y aplicar técnicas para el movimiento de cadenas cinemáticas, mismas que permitirán animar algunos de los movimientos que los seres humanos tienen como una capacidad natural.

El proyecto cuenta con un actor digital, el modelo de una mujer, el cual fue diseñado en la herramienta MilkshapeModel, su diseño cuenta con siete etapas que son: posicionar el esqueleto, dibujar bosquejo, dibujar puntos de referencia, insertar vértices, formar triángulos, agrupar triángulos y por último aplicar textura a grupos [1] y [4].

Una de las partes fundamentales del actor digital es el esqueleto, el cual esta conformado por huesos (eslabones) conectados a uniones o articulaciones, estas uniones están organizadas en un árbol [5]. La representación jerárquica de las uniones forman nodos padre y nodos hijo, de esta forma se permite heredar transformaciones. Por lo tanto, la herencia o el paso de información se maneja utilizando la teoría de cinemática directa⁵ [1].

⁵ Consiste en determinar cual es la posición y orientación del extremo final del robot, con respecto a un sistema de coordenadas que se toma como referencia.

Para la animación del actor digital fueron asignados vértices de triángulos para cada unión, por lo tanto el esqueleto del modelo está compuesto por 53 uniones, de las cuales a 14 se les aplicó cinemática directa (algoritmo Denavit-Hartenberg)[5].

Sobre las 14 articulaciones se aplicó la convención Denavit-Hartenberg, ésta permite mover una cadena articulada de acuerdo a cierto número de parámetros [1] y [4]. También se utilizó la articulación esfera-cavidad, la cual consiste en la unión de esfera-y-cavidad que posee tres grados de libertad rotatorios [4]. Ver Figura 1.2. Las articulaciones esfera-cavidad se utilizan para modelar articulaciones tales como el hombro y la cadera de los humanos. Para la animación de las 14 articulaciones, se definió un control en los grados de libertad de acuerdo a las características físicas del humano.

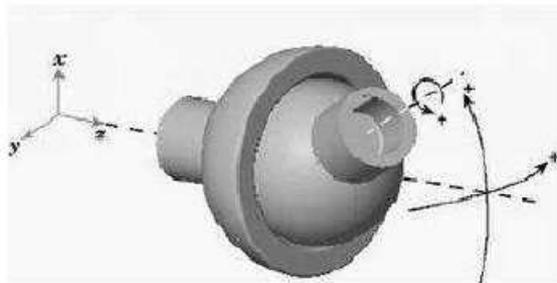


Figura 1.2 Articulación Esfera-y-cavidad

Dentro de las articulaciones controladas se tienen: dos en miembros superiores (hombro y codo) y dos en miembros inferiores (cadera y rodilla) con sus respectivos rangos de movilidad [4].

Para la graficación y demostración del movimiento del actor digital, se cuenta con un ambiente 3D simple, esta interfaz contiene elementos que permiten manipular al actor digital que son las flechas de teclado: arriba, abajo, izquierda, derecha (movimiento en un punto actor), las teclas PgUp- vista cámara desde arriba, pgDn- vista cámara desde abajo, i - movimiento izquierda actor, d - movimiento derecha y la animación de las articulaciones con las teclas 1 - 2 Rodilla , 3 - 4 Codo, 5 - 6 Cintura[1].

Por lo tanto, la razón por la que se realiza esta investigación, consiste aplicar inicialmente cinemática directa para simular los movimientos de una persona a través de un actor digital. La simulación modelará el movimiento de cada una de las restantes

articulaciones (39 articulaciones). Este movimiento debe ser lo más cercano al de un humano. Además de establecer una interfaz de software en la cual se deje de usar las teclas que manipulen las articulaciones por controles.

La aplicación de la cinemática directa sobre las articulaciones del actor digital, podrá permitir realizar aplicaciones basadas en simulación para el desarrollo futuro de un modelo de prótesis, la cual podrá recibir algún tipo de señal que le permita moverse. De esta manera el proyecto se puede extender para hacer modelos de prótesis para personas con diferentes tipos de discapacidad.

También mediante este tipo de actores, se busca simular situaciones de riesgo en las cuales esté en peligro la seguridad física del humano, como simuladores de colisión de autos o de trabajos peligrosos o inseguros.

Por otro lado, el actor digital será agregado en un trabajo futuro, a un ambiente 3D sofisticado, para aplicar algoritmos de planificación de movimientos que permitan dotarlo de cierto grado de inteligencia.

1.5. Cronograma de actividades

Tabla 1.1 Actividades para la propuesta de protocolo de investigación.

#	Actividad	Producto	Septiembre				Octubre				Noviembre				Diciembre			
			s1	s2	s3	s4	s1	s2	s3	s4	s1	s2	s3	s4	s1	s2	s3	s4
1	Revisión de literatura	Resumen de literaturas revisadas	■	■	■	■												
2	Elaboración de propuesta de protocolo de investigación	Documento de propuesta de protocolo de investigación				■	■	■	■									
3	Revisión de la primera versión del protocolo de investigación	Documento de protocolo de investigación								■								
4	Mejoras del protocolo de investigación (Marco Teórico)	Documento de Protocolo de investigación , marco teórico y diseño de la investigación									■	■	■	■				
5	Presentación de la propuesta de investigación	Diapositivas													■	■		
	s= semana																	

Tabla 1.2 Actividades para migrar el formato del modelo de un actor digital.

#	Actividad	Producto	Septiembre				Octubre			
			s1	s2	s3	s4	s1	s2	s3	s4
1	Revisión de la estructura para graficar el modelo del actor digital en el formato .ms3d	Documento y diagrama acerca de la grafición del actor digital								
2	Revisión de la función de lectura de la estructura del actor digital en el formato .ms3d	Documento y diagrama acerca de la lectura del actor digital								
3	Elaboración de una función de escritura de la estructura del actor digital en el formato .ms3d al formato .txt	Código de la función de escritura. (creación de 2 archivos datos.txt y datos2.txt)								
4	Comprobación del cambio de los datos al formato .txt	Documento de descripción de los archivos datos.txt y datos2.txt)								
5	Elaboración de una función de lectura de la estructura del actor digital en el formato .txt	Código de la función de lectura.(archivos datos.txt y datos2.txt)								
6	Graficación del actor digital de acuerdo a los datos del formato .txt	Documento de descripción de la graficación del actor digital								
	s= semana									

Tabla 1.3 Actividades para el estudio de la cinemática y articulaciones.

#	Actividad	Producto	Octubre				Noviembre				Diciembre						
			s1	s2	s3	s4	s1	s2	s3	s4	s1	s2	s3	s4			
1	Estudio y análisis de la teoría de la cinemática y la robótica	Documento de descripción de la teoría cinemática															
2	Análisis de la teoría de cinemática directa, uniones revolutiva y esfera cavidad en el cuerpo humano	Documento de descripción de la teoría cinemática directa y uniones															
3	Análisis de teoría de la convención de Denavit-Hartenberg	Documento de descripción de la teoría de convención de Denavit-Hartenberg															
4	Comprensión de la simbología de la convención de Denavit-Hartenberg y matriz de transformación	Documento de descripción de la simbología de la convención de Denavit-Hartenberg y matriz de transformación															
5	Aplicación de la cinemática directa en graficación 2D	Pruebas en una hoja de calculo de Excel															
6	Aplicación de la cinemática directa en graficación 3D	Pruebas en una hoja de calculo de Excel															
7	Asociación de las uniones revolutivas y cavidad esfera a las articulaciones del actor digital	Documento que describe que tipo de unión se usara para cada articulación del actor digital															
	s= semana																

Tabla 1.4 Actividades para codificar rutinas de cada una de las articulaciones del actor digital.

#	Actividad	Producto	Enero				Febrero				Marzo					
			s1	s2	s3	s4	s1	s2	s3	s4	s1	s2	s3	s4		
1	Revisión de la función kinema del actor digital	Documento y diagrama acerca de la función kinema.														
2	Análisis de las articulaciones y sus grados de libertad, además de definir las restricciones de movimiento en el ser humano	Documento que describe las articulaciones y sus grados de libertad y la especificación de los movimientos de un humano														
3	Análisis de las uniones del actor digital y su orientación	Prueba en una hoja de calculo de Excel (codo izquierda y rodilla izquierda) e imagen de la constitución del esqueleto y uniones del actor														
4	Elaboración de rutinas aplicando cinemática directa para mover las 39 articulaciones faltantes del actor digital	Código de 39 articulaciones.														
5	Comprobación de las rutinas para mover las 53 articulaciones del actor	Pruebas del movimiento de las 53 articulaciones.														
s= semana																

Tabla 1.5 Actividades para implementar la interfaz de controles y análisis de los algoritmos.

#	Actividad	Producto	Marzo				Abril				
			s1	s2	s3	s4	s1	s2	s3	s4	
1	Desarrollo de la interfaz de controles para cada articulación	Una interfaz de controles y documento de descripción de cada control									
2	Programación de cada uno de los controles para manipular cada una de las articulaciones del actor digital	Código de cada control									
3	Graficación del actor digital y manipulación de las articulaciones mediante los controles	Documento de pruebas de cada uno de los controles y el respectivo movimiento									
4	Análisis de algoritmos para lograr que camine el actor digital	Documento de descripción de los posibles algoritmos para hacer caminar al actor y especificación de ventajas y desventajas de estos algoritmos									
s= semana											

1.6. Alcances y limitaciones

Alcances

- Aplicar cinemática directa sobre cada articulación del actor digital respetando los grados de libertad de cada articulación.
- Desarrollar una ventana de controles que permita manipular los movimientos de las 53 articulaciones del actor digital.
- Analizar y diseñar un algoritmo para que el actor digital camine en un ambiente simulado.

Limitaciones

- Aplicar exclusivamente cinemática directa a cada una de las articulaciones del actor digital (no se implementará cinemática inversa).
- El algoritmo para que el actor camine requiere de mayor trabajo por lo que en este proyecto sólo se considera su análisis y diseño.

1.7 Recursos de hardware y software

Recursos de hardware

- Computadora con procesador AMD Athlon™ 64 X2 Dual-Core, con memoria de 1 G.
- Monitor predeterminado, tarjeta de video ATI Radeon Xpress 1150

Recursos de software

- Herramienta de desarrollo Borland C++ Builder 6.0.
- Herramienta para Graficación OpenGL API.
- Editor Dev C++ Versión 4.9.9.2
- Sistema Operativo Windows XP Profesional 5.1.2600 Service Pack 2
- Lenguaje de Programación C++.

Capítulo 2. Marco Teórico

2.1. Introducción a la cinemática

2.1.1. Definición de cinemática

Todo el entorno que conocemos se encuentra en constante movimiento. La cinemática que es la ciencia se encarga del estudio los movimientos de los objetos, para poder conocer su comportamiento y realizar estudios o aplicaciones de éstos en la vida diaria. La cinemática se define como la ciencia que estudia al movimiento la cual restringe una descripción geométrica, por medio de la posición, orientación, velocidad, tiempo y aceleración [11]. Por lo tanto, el término cinemática se refiere a la descripción matemática del movimiento sin considerar las fuerzas físicas subyacentes [1].

2.1.2. Definición de robótica

El concepto de robótica consiste en la combinación del campo de la ciencia de la computación y de la ingeniería para crear robots, dispositivos que pueden moverse y reaccionar con la entrada de datos provenientes de sensores [13].

La Asociación Francesa de Normalización (AFNOR) define los términos manipulador y robot de la siguiente manera:

Un *manipulador* es un mecanismo formado generalmente por elementos en serie, articulados entre sí, destinados al agarre y desplazamiento de objetos, es multifuncional y puede ser administrado directamente por un operador humano o mediante un dispositivo lógico.

Un *robot* es un manipulador automático servocontrolado, reprogramable, polivalente, capaz de posicionar y orientar piezas, materiales o dispositivos especiales, siguiendo trayectorias variables reprogramables, para la ejecución de tareas variadas. Su unidad de control incluye un dispositivo de memoria y ocasionalmente de percepción del entorno.

2.1.3 Cinemática del robot

La cinemática del robot estudia el movimiento del robot con respecto a un sistema de referencia. La cinemática se interesa por la descripción analítica del movimiento espacial del robot como una función del tiempo y en particular por las relaciones entre la posición y la orientación del extremo final del robot (localización), con los valores que toman sus coordenadas articulares [8]. El concepto de *cadena cinemática* consiste principalmente de una serie de elementos rígidos llamados eslabones (links) unidos mediante articulaciones (joints) que permiten un movimiento relativo entre dos eslabones consecutivos [17] y [18]. Este término también se le adjudica a un robot porque está constituido por cadenas cinemáticas. La Figura 2.1 ejemplifica una cadena cinemática.

La cadena cinemática está conformada por elementos tales como eslabones que tiene características físicas como el peso, material, inercia, tamaño, entre otras. Las articulaciones tienen la característica de *grados de libertad*.

Una cadena cinemática y principalmente en el actor digital con el que se cuenta está constituido por: A_1, A_2, \dots, A_n , un conjunto de n elementos. Para cada i tal que $1 \leq i < n$, el elemento A_i está unido al elemento A_{i+1} de tal modo que cualquier movimiento de A_{i+1} se realiza con respecto a A_i [1].

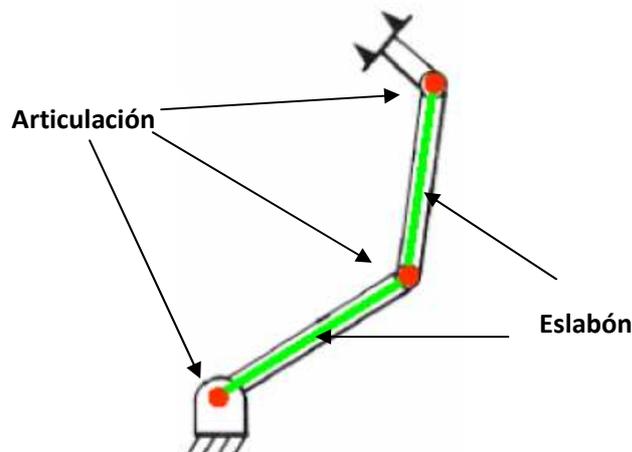


Figura 2.1 Representación de una cadena cinemática.

2.1.4. Cinemática directa y cinemática inversa

Existen dos problemas fundamentales para resolver la cinemática del robot: El primero es el problema *cinemático directo*, éste consiste en determinar cual es la posición y orientación del extremo final del robot, con respecto a un sistema de coordenadas que se toma como referencia, conocidos los valores de las articulaciones y los parámetros geométricos de los elementos del robot. El segundo problema denominado *cinemático inverso*, define la configuración que debe adoptar el robot para una posición y orientación del extremo conocidas [8] y [11]. La Figura 2.2 presenta la relación entre los dos problemas de la cinemática.

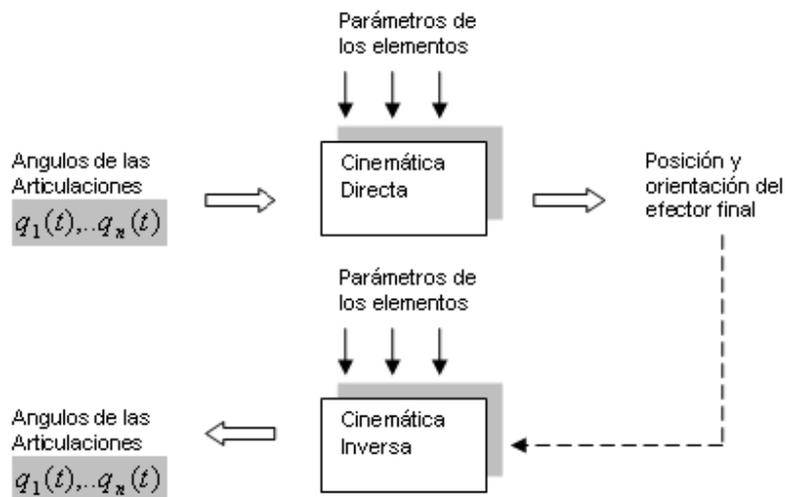


Figura 2.2 Relación entre la cinemática directa e inversa [1]

2.1.5. Transformaciones geométricas para mover cuerpos rígidos en 3D

En la graficación mediante computadora, se dibujan diferentes elementos como puntos, líneas, polígonos y variedades de objetos 3D. Desde un punto de vista técnico, los modelos u objetos 3D se consideran como un grupo de fórmulas matemáticas que describen un mundo en tres dimensiones.

Para la graficación y localización de los objetos es necesario determinar el objeto en un sistema de referencia (*Sistema Universal de Referencia*) [16]. Ver Figura 2.3.

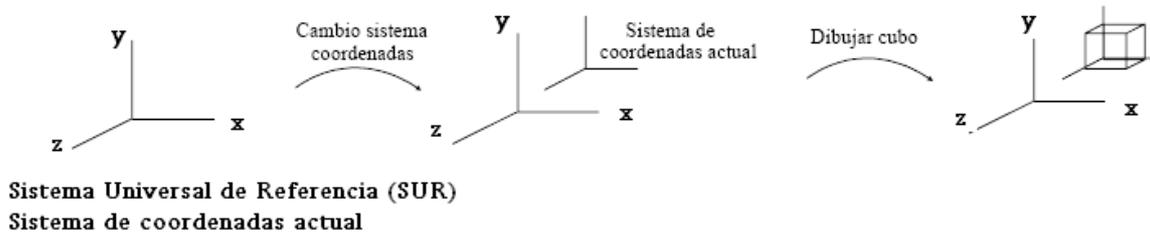


Figura 2.3 Representación de la graficación de un cubo en el Sistema Universal de Referencia.

El *Sistema Universal de Referencia* (SUR) es la base del mundo físico en tres dimensiones (3D), definido por un sistema referencial ortogonal formado por los ejes x , y , y z . Los tres ejes son mutuamente perpendiculares y se encuentran unidos en un punto común llamado *origen*, el cual se denota con 0. En un *sistema de referencia espacial*, un eje sirve como regla para medir la posición de un objeto teniendo un resultado en términos de metros o centímetros [19]. Al establecer el tamaño de los ejes en un *sistema de referencia espacial*, se establece un vector a cada eje, a este vector se le llama vector base. Por lo tanto, un *sistema coordenado cartesiano* es definido como un *sistema de referencia espacial* con vectores base en cada uno de sus ejes que son mutuamente perpendiculares [16] y [19]. Ver Figura 2.4.

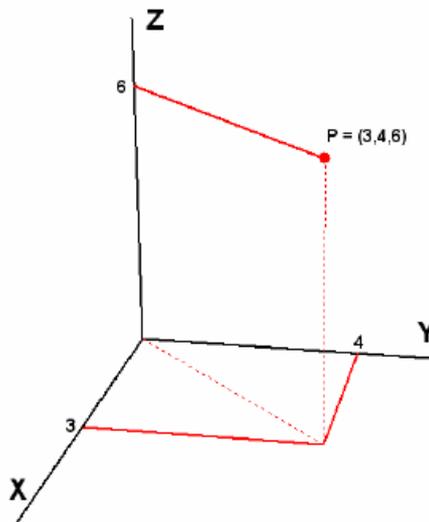


Figura 2.4. Sistema coordenado cartesiano

Para el modelado de los objetos, es necesario un mecanismo para transformar la posición, orientación y escala de los objetos de forma que se combinen en una escena o sistema de referencia común con otros elementos y realizar acciones sobre los objetos como mover o efectuar diversas copias del mismo en diferentes posiciones y tamaños.

Por lo tanto, al conjunto de modificaciones que se le aplican a un modelo u objeto en tres dimensiones (x, y, z), de modo que sea posible su representación en una pantalla bidimensional se le llama *transformación* [16].

Las operaciones básicas que realizan las *transformaciones* sobre los objetos se dividen en tres: la primera *rotaciones* al rededor de los ejes de coordenadas, la segunda *traslaciones* o cambios de posición y la tercera el *escalado* o cambios en las dimensiones de los objetos.

Cada vez que se aplique una operación de *transformación* a los objetos del sistema de referencia, es necesario modificar el sistema de coordenadas de los objetos. Como resultado, se obtendrá un nuevo sistema de coordenadas transformadas. Las coordenadas del sistema de referencia no cambian [16].

Para llevar a cabo la representación de figuras geométricas 3D y objetos 3D usando las operaciones básicas de transformación, es necesario el uso de herramientas matemáticas como lo son las matrices. A continuación se muestra la expresión general de una *matriz neta* (compuesta) 4x4, para realizar las tareas de *rotación*, *traslación*, *escalado* y la *proyección*.

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} & 0 \\ a_{21} & a_{22} & a_{23} & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ d_x & d_y & d_z & 1 \end{vmatrix}$$

Figura 2.5 Matriz neta modelo1

Definiendo la matriz neta de la Figura 2.5, la submatriz a_{ij} representa el cambio de *escala* y *rotación* neta, d_i define el vector de *traslación* neta de los puntos.

Si en la composición de la matriz se incluyera la proyección en perspectiva, la Figura 2.6 define a p_j como el vector de *perspectiva*.

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} & p_1 \\ a_{21} & a_{22} & a_{23} & p_2 \\ a_{31} & a_{32} & a_{33} & p_3 \\ d_x & d_y & d_z & 1 \end{vmatrix}$$

Figura 2.6 Matriz neta modelo 2

Las operaciones de transformación son siempre matriciales y pueden descomponerse en sumas y multiplicaciones. Para visualizar cómo se combinan las diferentes transformaciones, se puede utilizar una estructura de árbol como en la Figura 2.7 la cual especifica el orden de las operaciones de transformación y los objetos que afecta.

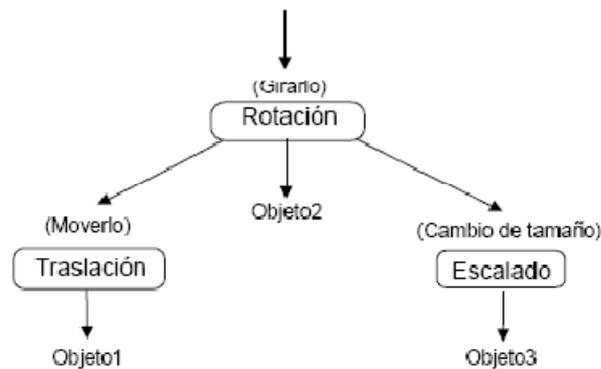


Figura 2.7. Operaciones de transformación [16]

La *transformación traslación* consiste en mover una cierta distancia un objeto en una dirección determinada, como consecuencia el objeto se encontrará en una posición nueva. Es decir los puntos (vértices) de un objeto son los que se trasladan de forma individual. Como el objeto está formado por polígonos y éstos a su vez por vértices, éstos últimos

cambian de posición y se redibujan formando nuevamente a los polígonos en la nueva ubicación. Ver Figura 2.8.

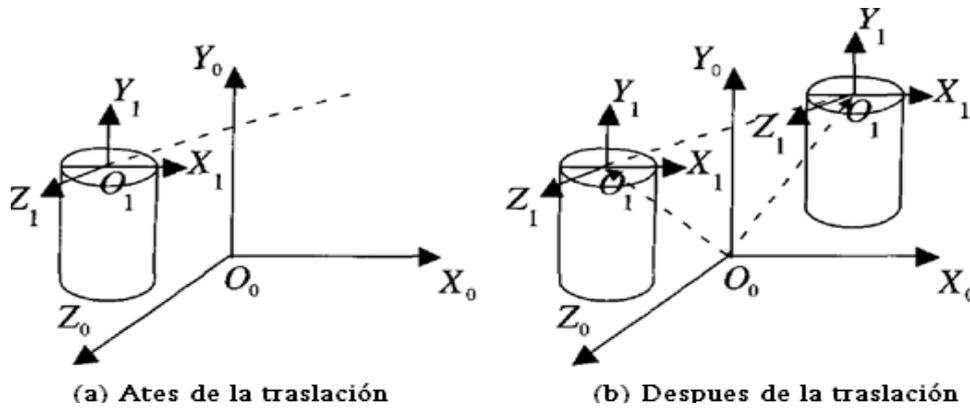


Figura 2.8 Representación de la transformación traslación de un cuerpo rígido [19]

Para la traslación en objetos 3D se define una matriz T 4×4 , con un vector de traslación $V=(x,y,z,1)$. Ver Figura 2.9.

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t_x & t_y & t_z & 1 \end{bmatrix}$$

Figura 2.9 Matriz de traslación modelo 1.

El vector de traslación queda después de esta transformación de esta manera $V' = (x', y', z', 1) = (x, y, z, 1) \cdot T$, siendo T la matriz de traslación. La expresión anterior es equivalente al sistema de ecuaciones.

$$\begin{aligned} x' &= x + t_x \\ y' &= y + t_y \\ z' &= z + t_z \\ 1 &= 1 \end{aligned}$$

La Figura 2.10 muestra un modelo de matriz de traslación que es utilizado para la cinemática directa.

$$\mathbf{T}(t_x, t_y, t_z) = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} x' \\ y' \\ z' \\ \omega' \end{pmatrix} = \mathbf{T}(t_x, t_y, t_z) \cdot \begin{pmatrix} x \\ y \\ z \\ \omega \end{pmatrix} = \begin{pmatrix} x + t_x \cdot \omega \\ y + t_y \cdot \omega \\ z + t_z \cdot \omega \\ \omega \end{pmatrix}$$

Matriz de traslación

Figura 2.10 Matriz de traslación modelo 2

Para realizar traslaciones inversas se aplica la matriz de traslación inversa denotada como \mathbf{T}^{-1} , solo se multiplica por -1 al vector de traslación. La Figura 2.11 representa la matriz inversa.

$$\mathbf{T}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -t_x & -t_y & -t_z & 1 \end{bmatrix}$$

Figura 2.11 Matriz de traslación inversa

La *transformación rotación*, consiste en el giro de los objetos. Para los objetos 3D, el desarrollador o usuario debe definir sobre que eje se desea rotar y también establecer el ángulo y el sentido (+ o -) de giro al rededor del eje seleccionado. Ver figura 2.12.

El giro puede ser relativo o absoluto, el primero consiste en que el objeto se gira con un ángulo θ sobre un eje dado, partiendo de la posición actual del objeto; el segundo define que si el objeto gira un ángulo θ a partir del estado cero (sistema de referencia).

La representación matricial de la rotación de un objeto 3D con respecto a un eje de coordenadas dependerá del eje utilizado la Figura 2.13 define las matrices de rotación correspondientes a cada uno de los ejes.

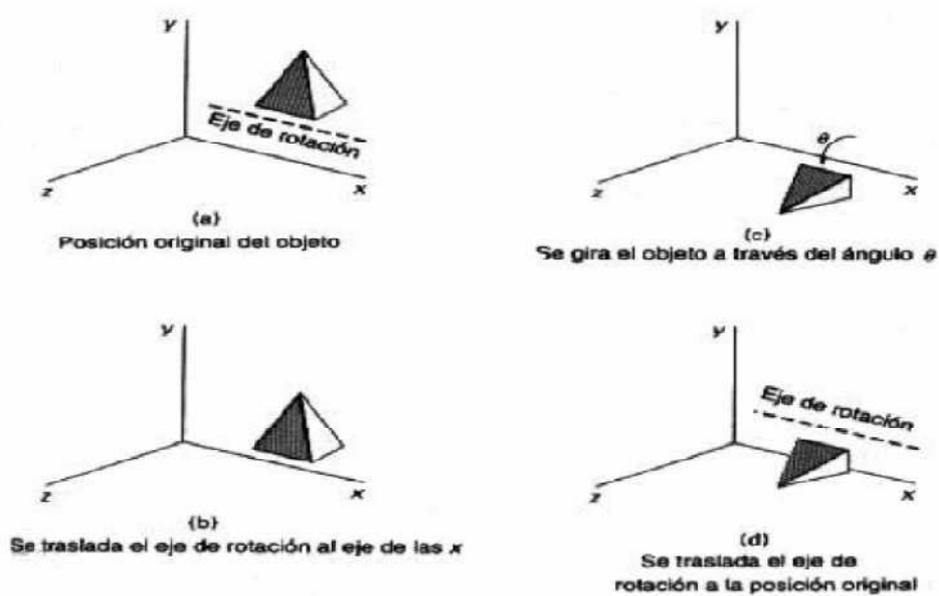


Figura 2.12 Giro de un objeto rígido 3D sobre un eje paralelo al eje x [16]

$$M_x(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & \sin \theta & 0 \\ 0 & -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \text{ Matriz de rotación alrededor del eje } x$$

$$M_y(\theta) = \begin{pmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \text{ Matriz de rotación alrededor del eje } y$$

$$M_z(\theta) = \begin{pmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \text{ Matriz de rotación alrededor del eje } z$$

Figura 2.13 Matrices de rotación para el cada eje

La *transformación escala* consiste en que los objetos 3D localizados dentro del espacio de referencia (Sistema Coordenado Cartesiano) puedan modificar su tamaño relativo en uno, dos o tres ejes. Para realizar esta operación se establece la matriz de escalado. Ver Figura 2.14.

$$\begin{pmatrix} x_2 \\ y_2 \\ z_2 \\ 1 \end{pmatrix} = \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{pmatrix}$$

Figura 2.14 Matriz de transformación escala

De la Figura 2.15 s_x, s_y, s_z , representan el grado en el que se aumentan las coordenadas del punto original con respecto a cada uno de los ejes.

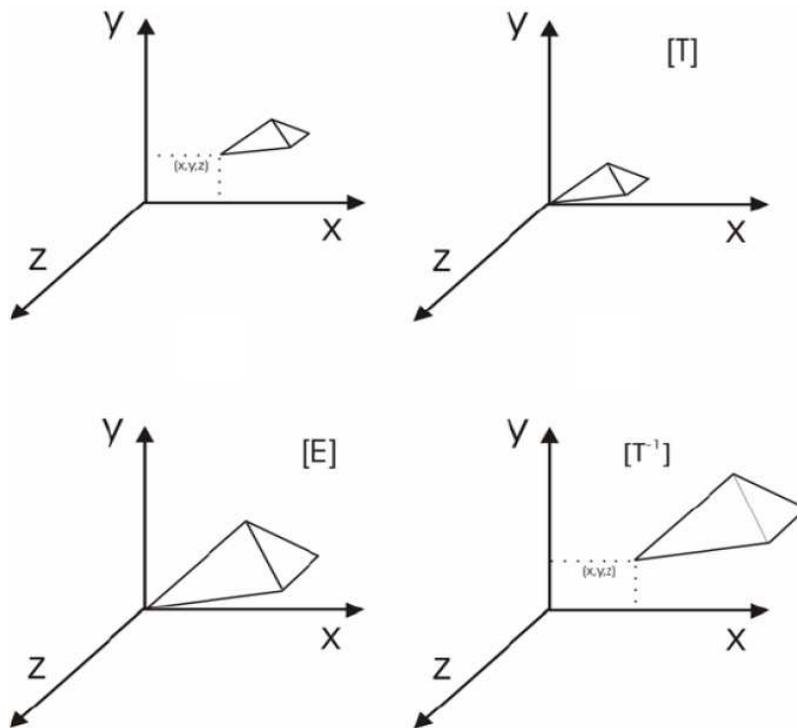


Figura 2.15. Escalado de un cuerpo rígido 3D [12]

Para poder realizar tanto rotaciones, translaciones, perspectiva y escalado en un espacio tridimensional, se define una sola *matriz de transformación homogénea* T_i de (4x4), esta matriz incluye las cuatro operaciones [8]. Y se representa en la Figura 2.16.

$$T = \left[\begin{array}{c|c} Rot(\phi, i_k) & p \\ \hline \eta & \sigma \end{array} \right]$$

Figura 2.16 Matriz de transformación homogénea

De la Figura 2.16 $Rot(\phi, i_k)$ es la matriz de rotación fundamental, p es el vector de translación, η es un vector de perspectiva, σ es un factor de escalamiento que usualmente se maneja igual a 1. De esta manera, cuando se realice una translación simple, la matriz de rotación fundamental se iguala a una matriz identidad, y cuando se trate de una rotación simple, el vector de translación se iguala a cero [8].

2.2. Cinemática

2.2.1. Uniones revolutivas y prismáticas

En robótica se hace uso típico de diversas articulaciones como son la *revolutiva* (rotatoria) R, la *prismática* (lineales) P. Una articulación revolutiva permite la rotación relativa entre dos elementos. Una articulación prismática permite el movimiento relativo lineal entre dos elementos [1] y [11].

Es importante mencionar que una articulación tiene grado de libertad (Degree of Freedom DOF), esto consiste en cada uno de los movimientos independientes que puede realizar cada articulación con respecto a la anterior. Las uniones revolutivas y prismáticas sólo cuentan con un sólo grado de libertad.

Cada articulación representa la interconexión entre dos elementos li y $li+1$. Se denota con zi al eje de rotación para una unión revolutiva o eje a lo largo del cual la unión prismática se mueve. Las variables de unión se denotan por θ_i para una articulación revolutiva y di para una articulación prismática [1] y [8]. Ver Figura 3.1.

El número de articulaciones representa el número de grados de libertad de un manipulador, típicamente un manipulador debe de estar constituido por lo menos de seis grados de libertad independientes: tres para posicionamiento y tres para orientación. Con menos de seis DOF el manipulador no puede alcanzar cada punto de orientación arbitrario en su ambiente de trabajo. El ambiente de trabajo es el área o espacio utilizado por el manipulador desde su primer elemento hasta el final [11].

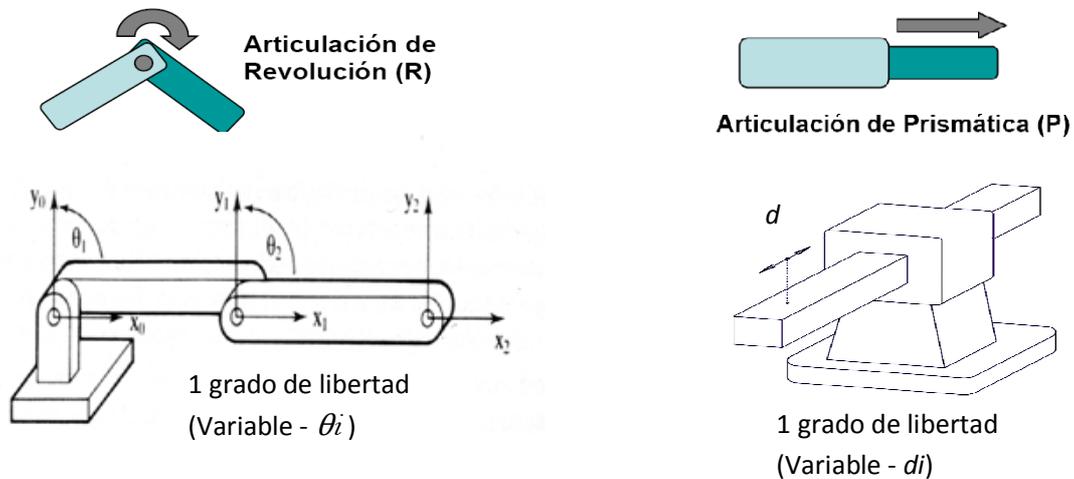


Figura 3.1 Articulación revolutiva y prismática

2.2.2. Convención Denavit-Hartenberg

Para describir la configuración (posición y orientación) de cada par de elementos consecutivo de una cadena cinemática, Denavit y Hartenberg propusieron un método matricial que establece de forma sistemática un sistema de coordenadas (sistema ligado al cuerpo) para cada elemento de una cadena cinemática [9].

El método Denavit-Hartenberg (DH) se basa en la caracterización de la configuración de un elemento A_i con respecto a un elemento A_{i-1} por medio de una matriz de transformación homogénea (4x4) que representa el sistema de coordenadas de cada elemento. Si cada par de elementos consecutivos está representado por su sistema de coordenadas asociado relacionado vía una matriz, entonces usando la multiplicación matricial de la cadena, es posible relacionar cualquiera de los elementos (ejemplo: una mano con respecto a otro elemento, el brazo) [1].

La representación Denavit-Hartenberg de un elemento rígido depende de cuatro parámetros geométricos asociados con cada elemento. Estos parámetros definen el paso de un sistema de referencia al siguiente sistema a través de una articulación. Los parámetros dependen de las características geométricas de cada elemento y de las articulaciones que les

unen con el anterior elemento y el siguiente (no dependen de la posición del robot) [1] y [9].

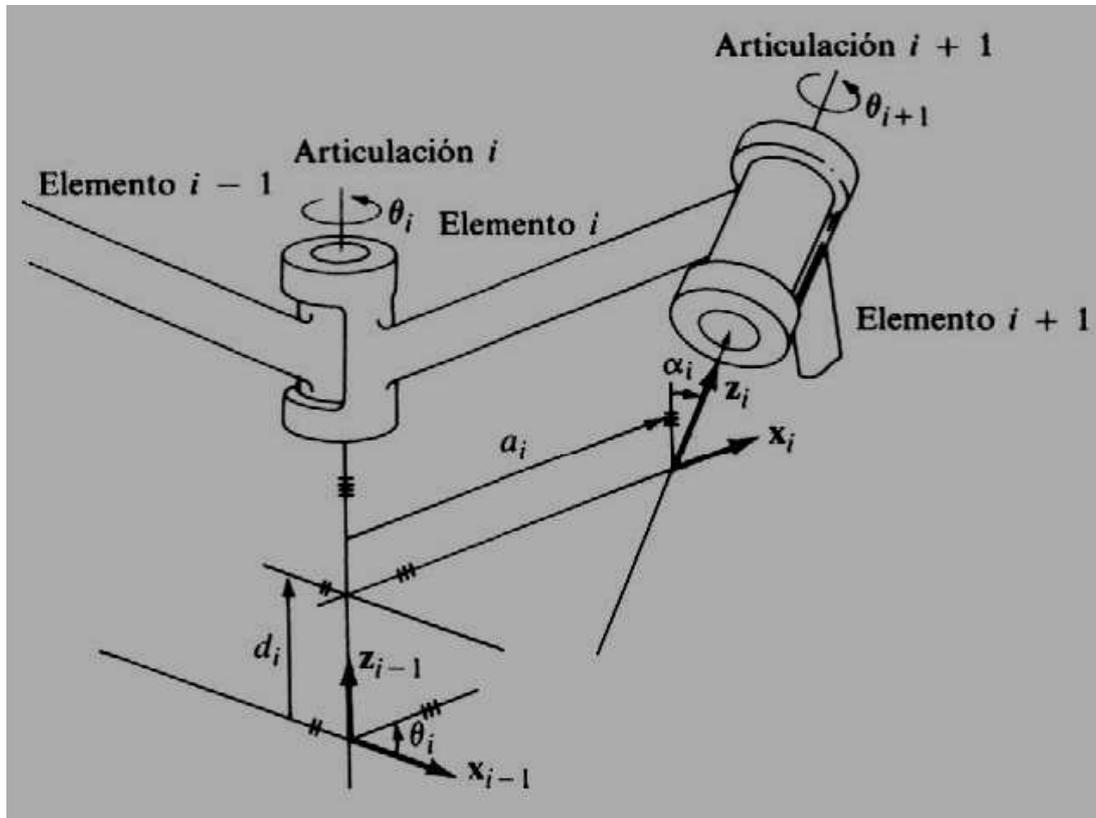


Figura 3.2 Representación de la convención Denavit-Hartenberg [9]

Los cuatro parámetros Denavit-Hartenberg se ilustran en la Figura 3.2. y estos se describen a continuación:

- 1.- θ_i : es el ángulo de la articulación del eje x_{i-1} al eje x_i respecto del eje z_{i-1} (utilizando la regla de la mano derecha). Ver Figura 3.3 b. Básicamente θ_i es el ángulo de rotación de un eslabón con respecto a otro, sobre el eje z_{i-1} .
- 2.- d_i : es la distancia desde el origen del sistema de coordenadas $(i-1)$ -ésimo hasta la intersección del eje z_{i-1} con el eje x_i a lo largo del eje z_{i-1} . Ver Figura 3.3 a; d_i es la distancia trasladada por un eslabón con respecto a otro a lo largo del eje z_{i-1} .
- 3.- a_i : es la distancia de separación desde la intersección del eje z_{i-1} con el eje x_i hasta el origen del sistema i -ésimo a lo largo del eje x_i para articulaciones revolutivas y para

articulaciones prismáticas se calcula como la distancia más corta entre los ejes z_{i-1} y z_i . Ver Figura 3.3 c. (Traslación a lo largo de x_i una distancia a_i).

4.- α_i : es el ángulo de separación del eje z_{i-1} al eje z_i respecto del eje x_i (utilizando la regla de la mano derecha). Ver Figura 1.14 d; α_i es ángulo de rotación al rededor del eje x_i .

Para una articulación revolutiva d_i , a_i y α_i son los parámetros de articulación y permanecen constantes para un robot, mientras que θ_i es la variable que cambia cuando el elemento A_i gira con respecto al elemento $A_i - 1$. Para una articulación prismática, θ_i , a_i y α_i son los parámetros de articulación y permanecen constantes para un robot, mientras que d_i es la variable de la articulación [14].

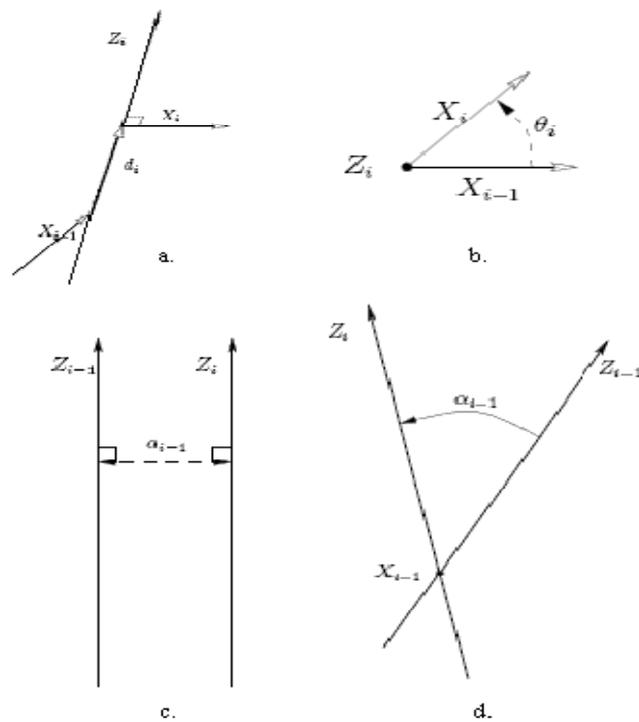


Figura 3.3 Parámetros de la convención Denavit-Hartenberg [1]

La matriz de *transformación homogénea* (4x4) utilizada para el método o convención DH, describe la *transformación* de un eslabón ($i-1$) a un eslabón i para una articulación i , ésta utiliza cuatro transformaciones básicas como se representa a continuación:

$$A_i = Rot_{z,\theta_i} Tras_{z,d_i} Tras_{x,a_i} Rot_{x,\alpha_i}$$

El producto de matrices es el siguiente:

$$= \begin{bmatrix} C_{\theta_i} & -S_{\theta_i} & 0 & 0 \\ S_{\theta_i} & C_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C_{\alpha_i} & -S_{\alpha_i} & 0 \\ 0 & S_{\alpha_i} & C_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} C_{\theta_i} & -S_{\theta_i}C_{\alpha_i} & S_{\theta_i}S_{\alpha_i} & a_iC_{\theta_i} \\ S_{\theta_i} & C_{\theta_i}C_{\alpha_i} & -C_{\theta_i}S_{\alpha_i} & a_iS_{\theta_i} \\ 0 & S_{\alpha_i} & C_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

donde C simboliza un coseno, y S se define como un seno.

Actualmente el método Denavit-Hartenberg se utiliza para el análisis y el control de manipuladores robóticos y en la coordinación del movimiento humano [1]. Por esta razón, el método se encuentra implementado en el actor digital para mover las articulaciones y se utilizará para dotar de movimiento a las 39 articulaciones de las 53 que tiene el actor.

Las matrices de transformación que se utilizara la definición matemática del actor digital [1] son:

Matriz ${}^{i-1}A_i$: *matriz de transformación homogénea* que representa la posición y orientación relativa entre los sistemas asociados a dos elementos consecutivos del robot, en donde ${}^{i-1}A$ se refiere al elemento relativo y A_i al elemento absoluto.

Conexión de matrices $A = {}^0A_3 = {}^0A_1 {}^1A_2 {}^2A_3$.

Matriz T: *matriz* 0A_n cuando se consideran todos los grados de libertad del robot,

$$T = {}^0A_6 = {}^0A_1 {}^1A_2 {}^2A_3 {}^3A_4 {}^4A_5 {}^5A_6.$$

2.2.3. Articulación esfera-y-cavidad

Una de las articulaciones más complejas es la articulación o unión esfera-y-cavidad, ésta representa articulaciones con tres grados de libertad rotatorios. Esta unión es compleja por las orientaciones 3D y la aplicación en articulaciones humanas [11].

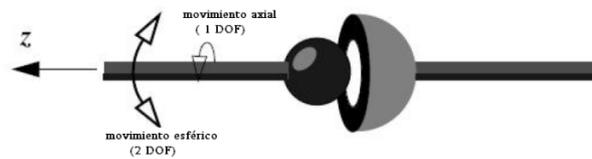


Figura 3.4 Unión esfera-y-cavidad [3]

La unión de esfera-y-cavidad posee tres grados de libertad rotatorios, se considera la más móvil de las articulaciones puramente rotatorias. Permite un movimiento axial o torcimiento, el cual tiene un grado de libertad como también un movimiento esférico u oscilación que determina su dirección. Este movimiento posee dos grados de libertad [1] y [2]. Ver Figura 3.4.

La articulación esfera-y-cavidad se utilizada para modelar articulaciones como el hombro y la cadera humanos. Ver Figura 3.5.

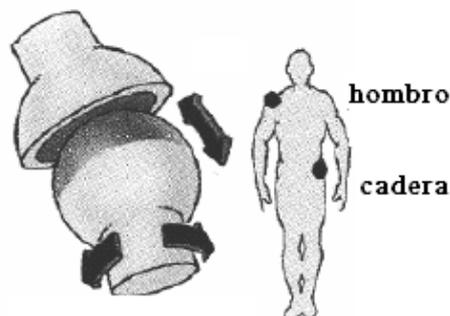


Figura 3.5 Implementación de la articulación esfera-cavidad en el humano

Matemáticamente, la representación de la unión esfera-y-cavidad consiste en una matriz local L que la Figura 3.6 define [5].

$$L_{R_{xyz}}(\theta_x, \theta_y, \theta_z) = \begin{bmatrix} C_y C_z & C_y S_z & -S_y & 0 \\ S_x S_y C_z & S_x S_y S_z + C_x C_z & S_x C_y & 0 \\ C_x S_y C_z + S_x S_z & C_x S_y S_z - S_x C_z & C_x C_y & 0 \\ T_X & T_Y & T_Z & 1 \end{bmatrix}$$

Figura 3.6 Matriz para la unión esfera-y-cavidad

De la Figura 3.6 x es el ángulo de rotación al rededor del eje X , y es el ángulo de rotación al rededor del eje Y , y z es el ángulo de rotación al rededor del eje Z . T_x indica la posición de una articulación en el eje X , T_y la posición en el eje Y , y T_z la posición en el eje Z . C representa coseno, y S seno.

2.3. Actor Digital

2.3.1. Definición de actor digital

Se le asigna el nombre de actor digital a un ente programado que existe y se mueve dentro de un mundo gráfico virtual, la generación de movimiento para un Actor digital es la base o la parte esencial para su control [4].

Otros términos usados para el actor digital son: humano virtual, agente animado, y hasta algunas veces Robot virtual (cuando su apariencia es robótica y no humana) [1]. La Figura 4.1 muestra algunos ejemplos de actores digitales.



Figura 4.1 Representación de actores digitales

El actor digital con el que cuenta el proyecto es un modelo de un humano (una mujer) la justificación del porque de una figura humana es con el fin de lograr la simulación de los movimientos de un humano por medio de la computadora y modelar en un futuro prótesis para personas con algún tipo de discapacidad.

También para simular situaciones de riesgo en las cuales esté en peligro la seguridad física del humano, como simuladores de colisión de autos o de trabajos peligrosos o inseguros.

2.3.2. El cuerpo humano y sus movimientos

En el diseño de modelos de seres humanos virtuales varía la noción de la *fidelidad virtual*, dependiendo de la aplicación que tendrá el actor digital (un videojuego, una película animada, material de enseñanza, entre otras). La fidelidad puede ser de apariencia (características físicas), de función (roles de una persona, límites de articulaciones), de tiempo (movimientos en tiempo real), de autonomía (toma de decisiones) y de individualidad (personalidad, rasgos psicológicos-fisiológicos) [1].

En el trabajo actual se busca la fidelidad de funcionamiento y como prioridad la del movimiento, esto consiste en que el actor digital tenga movimientos lo más cercano al de un humano. Por esa razón, es necesario conocer los principales rangos que tiene las articulaciones físicas de un humano para poder animar al actor digital y que éste se pueda comportar de acuerdo a los límites definidos por las características físicas.

La animación del actor digital se basa en los rangos de movimientos estándares en el cuerpo humano (Kinesiología), los cuales se muestran en la Figura 4.2, Figura 4.3, Figura 4.4, Figura 4.5, Figura 4.6 y Figura 4.7.

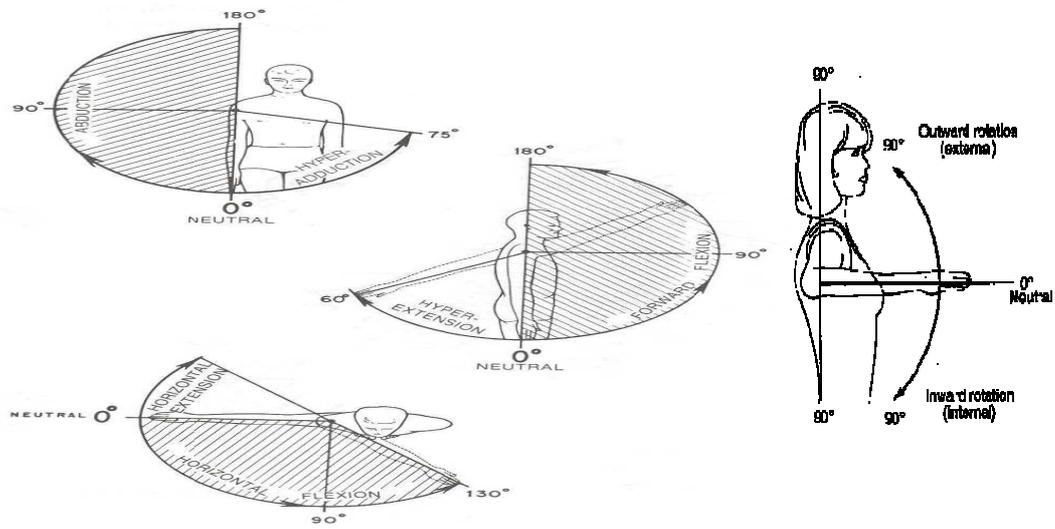


Figura 4.2 Rangos de movimiento del hombro [1] y [12]

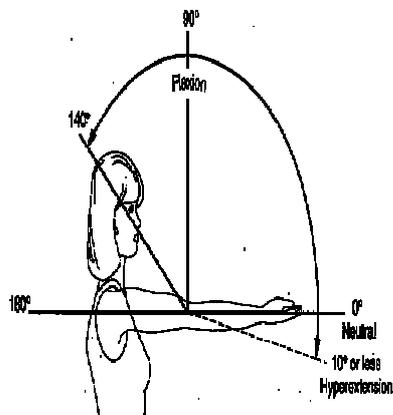


Figura 4.3 Rangos de movimiento del codo [1] y [12]

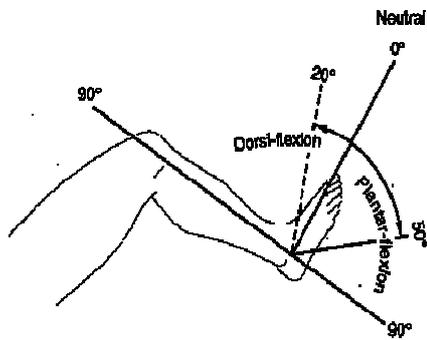


Figura 4.4 Rango de movimiento del tobillo [1] y [12]

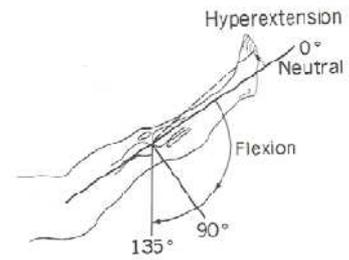


Figura 4.5 Rango de movimiento de la rodilla [1] y [12]

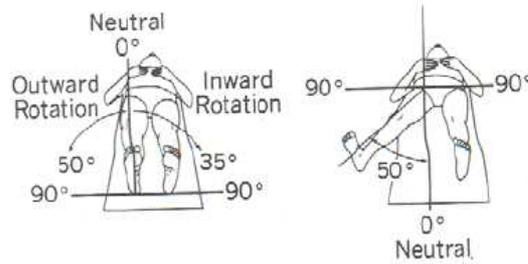
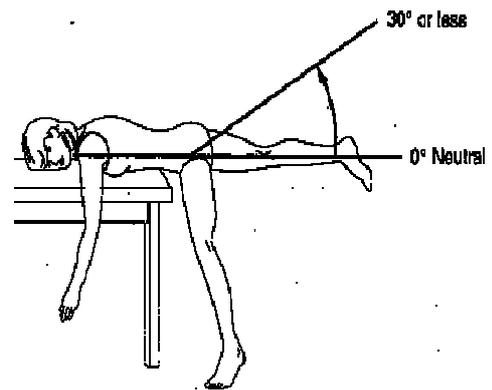
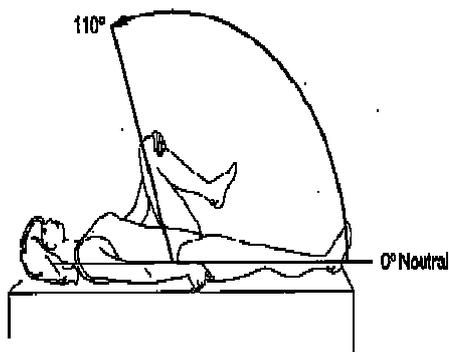


Figura 4.6 Rangos de movimiento de la pierna [1] y [12]

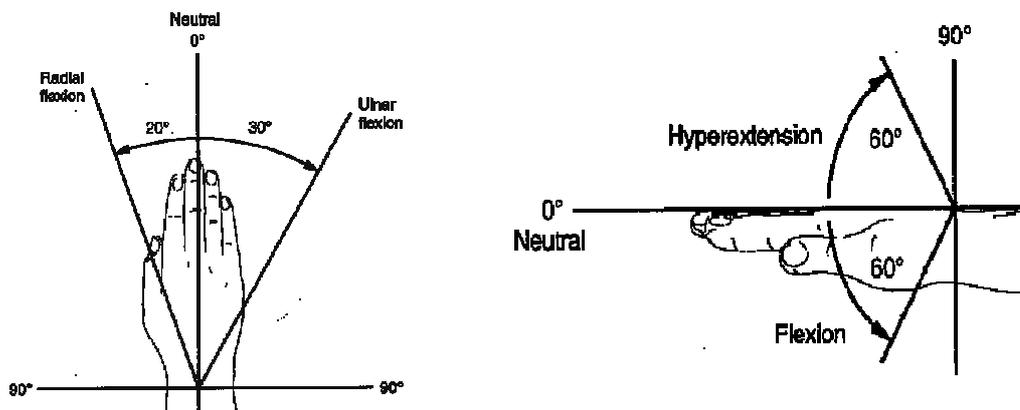


Figura 4.7 Rangos de movimiento de la muñeca [1] y [12]

A continuación las Tablas 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8 definen los grados de movimiento para algunas articulaciones del cuerpo humano.

Tabla 2.1 Rangos de articulación tobillo

Movimiento del tobillo	Grados
Dorsiflexión	20
Plantarflexión	50
Inversión	5
Eversión	5
Aducción hacia el pie	20
Abducción hacia el pie	12

Tabla 2.2 Rangos de articulación codo

Movimiento del codo (antebrazo)	Grados
Flexión	135
Extensión	-5
Protracción	90
Supinación	90
Transportar hombre	5
Transportar mujer	10 a 15

Tabla 2.3 Rangos de Cadera

Movimiento de la cadera	Grados
Anteversión normal	15
Flexión rodilla extendida	90
Flexión rodilla flexionada	120
Extensión	30
Abducción rodilla extendida	45 a 50
Abducción rodilla flexionada	70 a 75
Aducción	20 a 30
Rotación interna	35
Rotación externa(inclinación lateral)	45

Tabla 2.5 Rangos de articulación de dedo pulgar

Movimiento del dedo pulgar	Grados
Abducción	70
Flexión / extensión MCP	50
Flexión / extensión IP	-20 a 90

Tabla 2.4 Rangos de articulación rodilla

Movimiento de la rodilla	Grados
Flexión	130
Extensión	10
Rotación interna	10
Rotación externa	10

Tabla 2.6 Rangos de articulación dedos de la mano

Movimiento de los dedos de la mano	Grados
Flexión MCP	90
Flexión PIP	100
Flexión DIP	90
Extensión MCP	-30 a 45
Extensión PIP	0
Extensión DIP	0 a 10

Tabla 2.7 Rangos de articulación hombro

Movimientos del hombro	Grados
Abducción	180
Aducción	45
Flexión	150 a 170
Extensión	45
Protracción	20
Retracción	20
Brazo, codo flexionado en 90 grados	
Rotación interna	55
Rotación externa	45
Abducción del hombro en 90 grados	
Rotación interna	70
Rotación externa	70

Tabla 2.8 Rangos de articulación dedos del pie

Movimientos de los dedos del pie	Grados
Extensión del dedo grande	70 a 90
Flexión de los dedos (2-5)	45

El actor digital con el que se cuenta tiene controladas cuatro articulaciones: dos articulaciones en miembros superiores (hombro y codo) y dos articulaciones en miembros inferiores (cadera y rodilla) con los siguientes rangos [1] y [5].

Hombro:

- Aducción / h-aducción 255grados
- Flexión directa/ h-extensión 240 grados
- Extensión / flexión horizontal 180 grados

Codo:

- Flexión / h- extensión 160 grados

Cadera:

- Flexión / h – extensión 150 grados
- Aducción 100 grados
- Rotación / izq. Der. 180 grados

Rodilla:

- Flexión / h – extensión 135 grados

2.3.3. Modelo del actor digital

2.3.3.1. Modelado del actor digital

Un actor digital como ya se ha definido existe como un programa que se ejecuta en un ambiente virtual. Este ambiente es constantemente dibujado sobre un despliegue gráfico (frames) que producen una serie de imágenes, las cuales son interpretadas por el usuario como un aparente movimiento de los objetos [1].

Una de las tareas más complicada de realizar para los animadores es el modelado del cuerpo humano. Pero actualmente se cuenta con gran variedad de metodologías para modelado de humanos y realizar la captura, la digitalización de sus formas y tamaños existentes, las métricas, las técnicas y estándares [1] y [4].

El uso de herramientas ha ayudado a agilizar el trabajo de modelado de actores digitales. A continuación se describirán el proceso de modelado para el actor digital con el que el proyecto cuenta, por medio del uso de la herramienta de modelado 3D llamada Milkshape 3D [1] y [5]. Ver Figura 4.8.

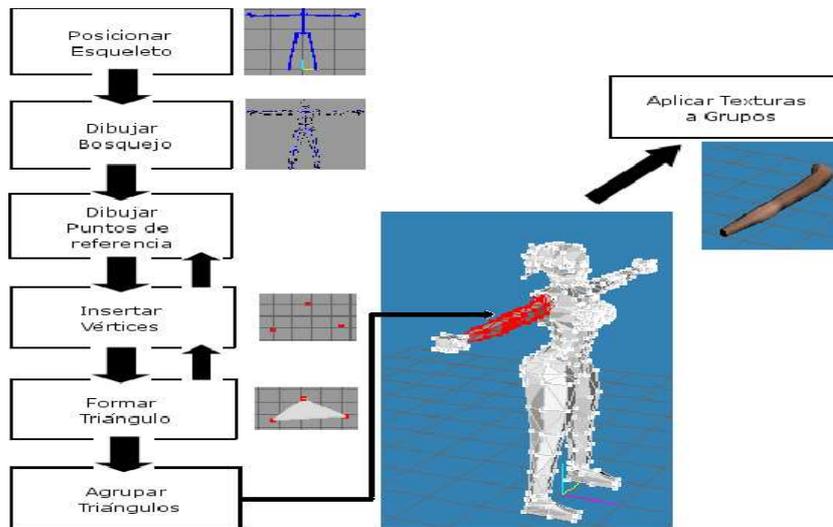


Figura 4.8 Diagrama del diseño del modelo [1]

La primera etapa del modelado del actor digital consiste en determinar las dimensiones del modelo, que es a través del esqueleto el cual se posiciono en el origen (x, y, z) en las coordenadas (0, 0, 0), la siguiente fase define la apariencia física que tendrá el actor, se diseña un bosquejo del modelo, el cual establece las características físicas del cuerpo humano. Terminado el bosquejo se le agregan vértices que sirven para crear triángulos. Estos triángulos se unen y forman grupos (mallas) de las diversas partes del actor digital que posteriormente se les aplica una textura (piel, vestuario) [1]. Ver Figura 4.14.



Figura 4.9 Modelo del actor digital [4]

2.3.3.2. El esqueleto del actor digital y sus articulaciones

A los animales con esqueleto interno se les llama vertebrados, y dentro de este grupo se encuentran seres humanos, mamíferos, reptiles, pescados, y pájaros. El esqueleto de un actor digital es análogo al esqueleto de un animal real, sin embargo los huesos digitales no tienen necesariamente que corresponder a los huesos reales [1] y [5].

El esqueleto del actor digital está compuesto por huesos (elementos o eslabones) conectados por uniones o articulaciones, que están organizados en un árbol como estructura de datos. La jerarquía del árbol consiste en seleccionar una unión o articulación como raíz en la cual se conectarán las otras uniones de manera jerárquica. Se puede elegir el nodo raíz de manera arbitraria, aunque es recomendable que esta unión se encuentre cerca del centro del actor digital.

La manera común para el caso de actores digitales es elegir la espina dorsal, para después poder unir la pelvis y el torso debajo de la raíz del árbol. Ver Figura 4.10.

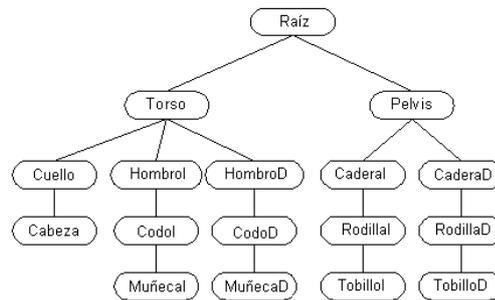


Figura 4.10 Jerarquía de las articulaciones del esqueleto

El esqueleto del modelo está compuesto por 53 uniones y por lo tanto se hace uso de esta teoría para definir un árbol que le permitirá su animación.

Después de la especificación del esqueleto del actor digital y del modelado se establecen los vértices para cada unión (joint), con el fin de preparar el modelo para su respectiva animación. De la debida asignación de vértices dependerá que el modelo no se deforme al momento de realizar algún movimiento. La Figura 4.17 representa el esqueleto y la malla de triángulos del actor digital.

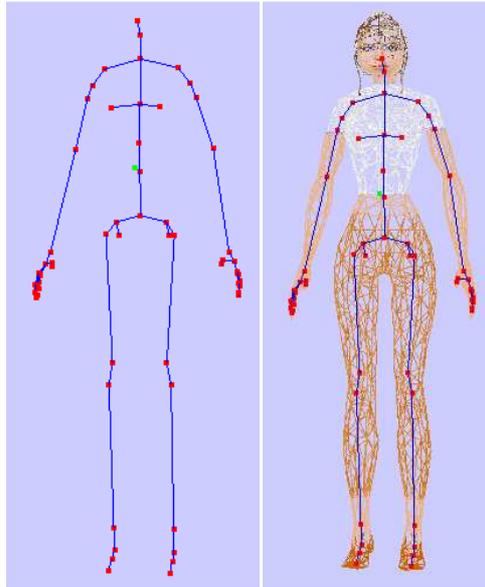


Figura 4.11 Esqueleto y malla de triángulos del actor digital

Por lo tanto, el esqueleto se puede usar como una armadura invisible que sirve como marco de referencia para posicionar y orientar la estructura geométrica del actor y también para definir el cuerpo, el movimiento y la personalidad del actor digital [1].

2.4. Trabajo relacionado

Como trabajo relacionado,[1] y [5] proponen un actor digital, el cual fue modelado en la herramienta Milkshape 3D. Este modelo tiene un formato .ms3d, y consta de una estructura de árbol en el cual se encuentran los nodos (articulaciones) que permiten el movimiento del actor. El actor digital consta de 53 articulaciones de las cuales 14 contienen movimiento. El movimiento tuvo la implementación de la teoría de la cinemática directa, con la modificación del algoritmo de Denavit-Hartenberg (matriz de transformación) y la esfera-y-cavidad (matriz esfera-y-cavidad). Para la animación de las 14 articulaciones del actor se define un control en los grados de libertad de acuerdo a las características físicas del humano.

[1] y [5] lograron el control de dos articulaciones en miembros superiores (hombro y codo) y dos articulaciones en miembros inferiores (cadera y rodilla) con sus respectivos rangos de movilidad y para la graficación y demostración del movimiento del Actor digital desarrollaron un ambiente 3D simple. Ver Figura 5.1.



Figura 5.1 Actor digital propuesto por [1] y [5]

Por lo tanto, en el trabajo actual utilizaremos el actor digital propuesto por [1] y [5], pero con la diferencia que la nueva forma de graficar el modelo es con la generación de un formato .txt con el fin de eliminar el formato binario .ms3d del actor y de un buffer para cargar los datos del actor digital, este nuevo formato permitirá almacenar los datos del

actor de una forma comprensible para el desarrollador y una lectura de los datos rápida hacia las estructuras del actor. También se aplicará cinemática directa sobre las 39 articulaciones restantes, estableciendo el control de los movimientos de acuerdo a las características físicas del humano para producir un movimiento real.

Por otra parte, [2] desarrollaron un planificador de movimientos para un maniquí digital (actor digital Eugene) el cual interactúa en un ambiente 3D cerrado. El maniquí propuesto por [2] contiene un movimiento integral y de planificación aproximada. Para lograr estos resultados su trabajo consistió en tres etapas: la primera evita obstáculos 3D, la segunda movimiento viable (permitir que el actor se desplace), la tercera manipula objetos. Esto fue conseguido por medio de la combinación de tres técnicas: métodos de planificación de movimientos, captura de movimiento basado en el control para caminar (animación del maniquí) y por último técnicas de cinemática inversa para la manipulación de objetos. Ver Figura 5.2.

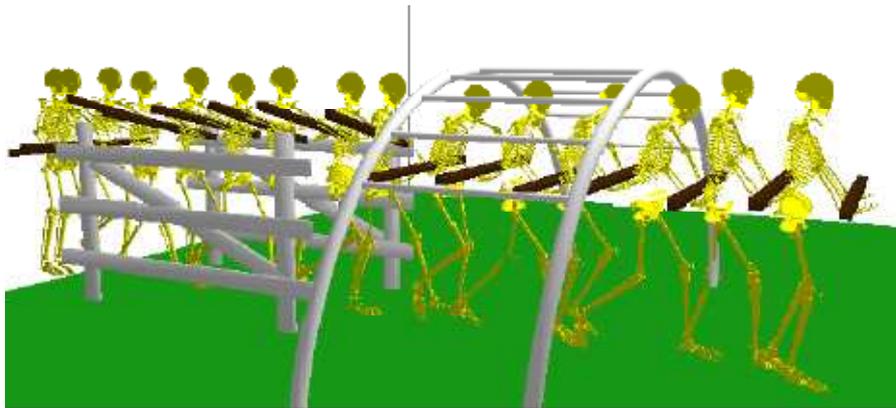


Figura 5.2. Actor digital Eugene propuesto por [2]

En este trabajo se realizará un análisis de algoritmos y técnicas para el control de movimiento para permitir que el actor digital camine, esto consiste en que realice un desplazamiento dentro de un ambiente 3D simple. Aun no se contempla la planificación de movimiento. Cabe mencionar que la investigación propuesta por [2] define un actor digital que manipula un objeto en las manos (barra de metal) y que se encuentra dentro de un ambiente cerrado 3D. Por lo tanto el actor digital por medio de la planificación tiene un punto inicial y final, esto conlleva a que realiza un desplazamiento (animación para caminar).

Capítulo 3. Diseño de la investigación

3.1. Introducción

La simulación de humanos por medio de la computadora a través de actores digitales ha permitido conocer el comportamiento que tendrá un robot humano físico en el futuro.

El objetivo del proyecto presente consiste en aplicar cinemática directa a las 39 articulaciones de las 53 de un actor digital por medio de una interfaz de software, con controles que permitan manipular cada una de las articulaciones. También se realiza un análisis y diseño de un algoritmo para que el actor digital pueda caminar.

En este capítulo se describe el diseño de la interfaz de software que permitirá controlar cada una de las articulaciones del actor digital y la graficación del movimiento de la articulación de acuerdo al control seleccionado, el desarrollo es bajo el lenguaje C++ y el API OpenGL y Windows.

La interfaz de software consiste de una área donde se grafica el actor digital y se observa el movimiento de las articulaciones; un área de controles con pestañas para las diferentes partes del cuerpo, por ejemplo, brazos, piernas, manos, pies, tórax, cabeza, entre otras, donde cada pestaña presenta las articulaciones respectivas para el área del cuerpo seleccionada. Un usuario manipula una articulación a la vez por medio de una barra de desplazamiento. Se puede observar mediante una caja de texto los valores que toma la articulación al manipular la barra de desplazamiento y en el área de graficación el movimiento de la articulación seleccionada. La interfaz desarrollada se ilustra en la Figura 6.1.

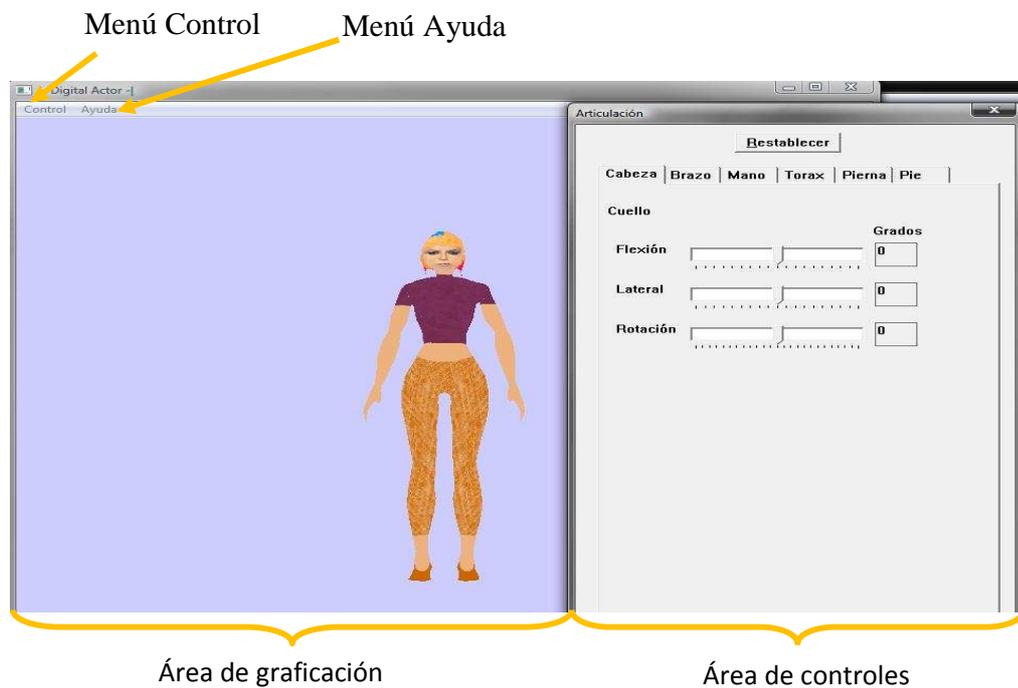


Figura 6.1 Pantalla de la aplicación *Digital Actor*

A continuación se definen los requerimientos funcionales y no funcionales de la aplicación.

3.2. Requerimientos

Requerimientos funcionales

- Leer modelo de un actor digital en formato .txt
- Graficar modelo de un actor digital
- Rotar y trasladar el modelo de un actor digital en ejes x,y, (implementado)
- Graficar modo malla y modo esqueleto el modelo de un actor digital (implementado)
- Implementar Luces y graficar del sistema cartesiano (ejes x,y,z) (implementado)
- Controlar y mover las 53 articulaciones
- Graficar el movimiento de un actor digital de acuerdo a la articulación seleccionada en el área de controles.

Requerimientos no funcionales.

- **Adaptabilidad:** La aplicación será capaz de seguir funcionando al agregar nuevas funciones o rutinas en lenguaje C++ para aplicarlas al actor digital.
- **Interfaz:** El uso de pestañas que definen la parte del cuerpo y que contienen las respectivas articulaciones (con sus barras de desplazamiento) que puede manipular el usuario, permitirá una mejor comprensión del actor digital y de un humano.
- **Extensibilidad:** La aplicación podrá migrarse al sistema operativo Linux.
- **Precisión:** La aplicación establecerá la exactitud en cuestión de los rangos de movimiento de cada articulación de un actor digital de acuerdo a las características físicas de un humano
- **Documentación:** Se realizara un manual de usuario sobre la interfaz de software para una mayor comprensión.

3.3. Documento de requerimientos

Tabla 3.1 Requerimiento lectura del modelo de un actor digital en formato .txt

Nombre de requerimiento:	Lectura del modelo de un actor digital en formato .txt
Descripción corta:	Realiza la lectura de un actor digital en formato .txt y la graficación del actor.
Descripción detallada:	1.- El usuario ejecuta el programa (<i>Digital Actor</i>) y este mostrara la ventana con el actor digital graficado. 2.- El usuario no tiene la opción de leer su propio actor digital, ya existe uno establecido.

Tabla 3.2 Requerimiento Control y movimiento de 53 articulaciones.

Nombre de requerimiento:	Control y movimiento de las 53 articulaciones del actor digital.
Descripción corta:	Representar gráficamente el control y el movimiento de las articulaciones del actor digital.
Descripción detallada:	1. El usuario controlara las 53 articulaciones del actor digital por medio del área de controles. 2. El usuario hace uso de un control (barra de desplazamiento) de una articulación y podrá manipularlo hacia la izquierda y derecha y visualizar el movimiento graficado del actor digital y sus rangos.

3.4. Casos de uso y casos de prueba

El usuario será capaz de realizar diferentes actividades como se muestra en la Figura 6.2.

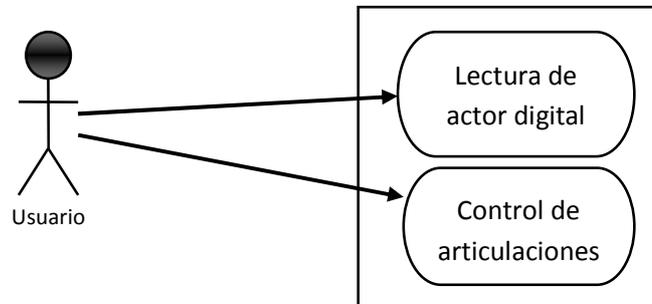


Figura 6.2 Casos de uso

A continuación se definen algunos casos de usos (C.U) y casos de prueba (C.P) para la aplicación del actor digital.

Caso de uso lectura de actor digital

Nombre	C.U. Lectura de un modelo de actor digital
Actor principal	Usuario
Precondiciones:	<ol style="list-style-type: none"> 1. El formato del modelo del actor digital debe ser .txt 2. El usuario no puede leer su propio modelo de actor digital.
Descripción	En este C.U. se leen los datos de una actor digital desde dos archivos de texto con extensión .txt
Datos de entrada	<p>Dos archivos de texto:</p> <p>datos.txt (contiene datos de vértices y de triángulos)</p> <p>datos 2.txt (contiene datos de grupos, materiales y joints)</p>
Datos de salida	Actor digital graficado o un mensaje de fallo de lectura de archivos

Escenario principal de éxito	<ol style="list-style-type: none"> 1. El usuario ejecuta el programa digital actor. 2. Visualizará el actor digital.
Extensiones o flujos alternativos	<p>Fallo de lectura</p> <ol style="list-style-type: none"> a) Ejecutar nuevamente la aplicación digital actor b) Verificar que existan los dos archivos .txt

Caso de prueba (C.P)

Nombre	C.P Lectura de un modelo de actor digital para el C.U. lectura de un modelo de actor digital
Datos de entrada	<p>datos.txt (contiene datos de vértices y de triángulos)</p> <p>datos 2.txt (contiene datos de grupos, materiales y joints)</p>
Datos de salida	Si la lectura de los archivos es correcta se grafica el actor digital, de lo contrario muestra el siguiente mensaje de error: “Error de lectura de los archivos datos.txt y datos2.txt”.

Casos de usos control de articulaciones.

Nombre	C.U. Movimiento de la articulación codo derecho de un actor digital
Actor principal	Usuario
Precondiciones:	<ol style="list-style-type: none"> 1. Ejecutar programa <i>Digital Actor</i> 2. Dirigirse ala parte de controles de la ventana.
Descripción	En este C.U. se elige la pestaña brazos y dentro de ella seleccionar

	la barra de desplazamiento con la etiqueta “codo derecho” de un actor digital para mover esta articulación.
Datos de entrada	Desplazamiento de la barra que contiene rangos exclusivos de la articulación codo derecho.
Datos de salida	Graficación de la nueva posición del codo derecho y muestra de los grados que incrementa o reduce en una caja de texto por el desplazamiento de la barra.
Escenario principal de éxito	<ol style="list-style-type: none"> 1. El usuario manipula la barra de control de la articulación codo derecho 2. Visualizará el movimiento del actor digital.
Extensiones o flujos alternativos	<p>Fallo de movimiento</p> <ol style="list-style-type: none"> a) Ejecutar nuevamente la aplicación Digital Actor

Caso de prueba (C.P)

Nombre	C.P Movimiento de la articulación codo derecho para el C.U. movimiento de la articulación codo derecho
Datos de entrada	Valor del grado de la articulación codo derecho
Datos de salida	Graficación de la nueva posición del codo derecho y muestra de los grados que incrementa o reduce en una caja de texto.

Caso de uso (C.U)

Nombre	C.U. Movimiento de la articulación codo izquierdo de un actor digital
Actor principal	Usuario
Precondiciones:	1. Ejecutar programa digital actor 2. Dirigirse a la parte de controles de la ventana.
Descripción	En este C.U. se elige la pestaña brazos y dentro de ella seleccionar la barra de desplazamiento con la etiqueta “codo izquierdo” de un actor digital para mover esta articulación.
Datos de entrada	Desplazamiento de la barra que contiene rangos exclusivos de la articulación codo izquierdo.
Datos de salida	Graficación de la nueva posición del codo izquierdo y muestra de los grados que incrementa o reduce en una caja de texto por el desplazamiento de la barra.
Escenario principal de éxito	1. El usuario manipula la barra de control de la articulación codo izquierdo. 2. Visualizará el movimiento del actor digital.
Extensiones o flujos alternativos	Fallo de movimiento a) Ejecutar nuevamente la aplicación digital actor

Caso de prueba (C.P)

Nombre	C.P Movimiento de la articulación codo izquierdo para el C.U. movimiento de la articulación codo izquierdo.
Datos de entrada	Valor del grado de la articulación codo izquierdo
Datos de salida	Graficación de la nueva posición del codo izquierdo y muestra de los grados que incrementa o reduce en una caja de texto.

Caso de uso (C.U)

Nombre	C.U. Movimiento de la articulación rodilla derecha de un actor digital
Actor principal	Usuario
Precondiciones:	1.Ejecutar programa digital actor 2. Dirigirse ala parte de controles de la ventana.
Descripción	En este C.U. se elige la pestaña piernas y dentro de ella seleccionar la barra de desplazamiento con la etiqueta “rodilla derecha” de un actor digital para mover esta articulación.
Datos de entrada	Desplazamiento de la barra que contiene rangos exclusivos de la articulación rodilla derecha.
Datos de salida	Graficación de la nueva posición de la rodilla derecha y muestra de los grados que incrementa o reduce en una caja de texto por el desplazamiento de la barra.
Escenario principal de	1. El usuario manipula la barra de control de la articulación rodilla derecha.

éxito	2. Visualizará el movimiento del actor digital.
Extensiones o flujos alternativos	Fallo de movimiento a) Ejecutar nuevamente la aplicación digital actor

Caso de prueba (C.P)

Nombre	C.P Movimiento de la articulación rodilla derecha para el C.U. movimiento de la articulación rodilla derecha.
Datos de entrada	Valor del grado de la articulación rodilla derecha
Datos de salida	Graficación de la nueva posición de la rodilla derecha y muestra de los grados que incrementa o reduce en una caja de texto.

Caso de uso (C.U)

Nombre	C.U Movimiento de la articulación rodilla derecha de un actor digital
Actor principal	Usuario
Precondiciones:	1. Ejecutar programa digital actor 2. Dirigirse ala parte de controles de la ventana.
Descripción	En este C.U. se elige la pestaña piernas y dentro de ella seleccionar la barra de desplazamiento con la etiqueta “rodilla izquierda“ de un actor digital para mover esta articulación.
Datos de entrada	Desplazamiento de la barra que contiene rangos exclusivos de la

	articulación rodilla izquierda.
Datos de salida	Graficación de la nueva posición de la rodilla izquierda y muestra de los grados que incrementa o reduce en una caja de texto por el desplazamiento de la barra.
Escenario principal de éxito	1. El usuario manipula la barra de control de la articulación rodilla izquierda. 2. Visualizará el movimiento del actor digital.
Extensiones o flujos alternativos	Fallo de movimiento a) Ejecutar nuevamente la aplicación digital actor

Caso de prueba (C.P)

Nombre	C.P Movimiento de la articulación rodilla izquierda para el C.U. movimiento de la articulación rodilla izquierda.
Datos de entrada	Valor del grado de la articulación rodilla izquierda
Datos de salida	Graficación de la nueva posición de la rodilla izquierda y muestra de los grados que incrementa o reduce en una caja de texto.

*Para las restantes articulaciones se utiliza el mismo modelo de caso de uso y caso de prueba, por lo tanto no es necesario mencionarlos.

Tabla 3.3 Especificación de actores

Nombre del actor:	Usuario
Descripción:	Persona que ejecuta el programa <i>Digital Actor</i>
Características:	Usuario que tenga relación con la robótica
Relaciones:	Interactúa con la aplicación para manipular las 53 articulaciones del actor digital, permitiéndole conocer el movimiento de un robot físico con forma humana.

3.5. Módulo

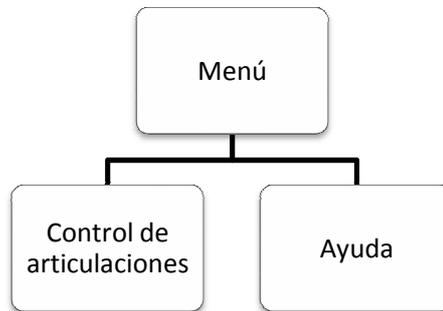


Figura 6.3 Módulos de la aplicación Digital Actor

Tabla 3.4 Módulo Ayuda

Nombre del módulo:	Ayuda
Descripción:	Contiene las instrucciones de uso de la aplicación <i>Digital Actor</i> e información sobre la versión de la aplicación.
Datos de entrada:	Solo dar clic en menú, “ayuda”, “ayuda digital actor” o “acerca de digital actor”.
Datos de salida:	Ventana con la información correspondiente a la opción seleccionada (instrucciones de uso o acerca de la aplicación).

Tabla 3.5 Módulo Control de articulaciones

Nombre del módulo:	Control de articulaciones
Descripción:	Contiene todos los controles para permitir el movimiento de las 53 articulaciones divididas en hojas de acuerdo a la parte del cuerpo a mover.
Datos de entrada:	Selección de un control (barra de desplazamiento) con este permite manipular los rangos de una articulación y de esta forma hacer cálculos para graficar.
Datos de salida:	Graficación del actor digital y del movimiento de la articulación que se está manipulando.

3.6. Manejo de datos

Los datos que se utilizarán están relacionados con las 53 articulaciones (joints) de un actor digital, estos se encuentran almacenados en un archivo de texto llamado datos2.txt. Los datos de las articulaciones que se encuentran almacenados son los siguientes: número de joint, nombre del joint, nombre del joint padre, número del joint padre, 3 coordenadas de traslación, 3 coordenadas de rotación.

Además de contener los datos de los joints, el archivo datos2.txt tiene almacenado los datos de grupos (conjunto de triángulos que conforman las mallas del modelo del actor digital) y materiales (datos de las texturas que son aplicadas a las mayas).

La aplicación realizará la lectura de los archivos datos.txt (contiene datos de vértices y de triángulos) y datos2.txt, y organiza las 53 articulaciones en un árbol binario (establece una jerarquía de acuerdo a los joints padres) para poder realizar la graficación del modelo del actor digital.

Para la manipulación de los datos se encuentran dos usuarios: el desarrollador y el usuario final, el primero realiza la lectura de los datos y la manipulación de las rutinas del programa así como las pruebas mediante la interfaz de software. El segundo usuario sólo hace uso de la interfaz de software, la cual le permitirá visualizar los datos mediante la graficación del actor digital y podrá manipular los datos de las articulaciones mediante controles (barras de desplazamiento) que permitan el movimiento.

Los datos de los archivos datos.txt y datos2.txt no sufren cambios por ningún usuario ya que solamente son leídos y cargados en las estructuras de la aplicación, quien se encarga de realizar las debidas transformaciones para graficar al actor digital y permitir el movimiento.

Por lo tanto, no existirá actualización de los datos que contengan los archivos con extensión .txt ni mantenimiento, tampoco hay necesidad de distribuir los datos. Otro punto importante es que no existe costo por emplear el modelo del actor digital (datos), puesto que corresponde a un trabajo de tesis [1].

Como ya se mencionó se aplicará cinemática directa a 39 articulaciones de las 53 con las que cuenta el actor digital, y para ello se necesita conocer los datos de los 53 joints para graficar debidamente los movimientos con respecto a los rangos de movimiento de un humano.

Las pruebas de los movimientos de las 53 articulaciones se realizarán mediante la interfaz de software. Para cada articulación hace uso de su respectiva barra de desplazamiento y se verificará que la articulación coincida con el movimiento y que el rango de movimiento sea respetado. Además de que la graficación del movimiento vaya de acuerdo a los datos de entrada.

Capítulo 4. Implementación

4.1. Manipulación de archivos .txt

El modelo del actor digital (figura de una mujer) con el que se cuenta fue diseñado sobre la herramienta MilkshapeModel y almacenado con el formato binario .ms3d [1], que al ser leído por la aplicación *Digital Actor* tenía que ser cargado a un buffer y posteriormente a las estructuras del actor digital implementadas por [1].

Para reducir el uso del buffer que requiere el formato .ms3d, el proyecto propone la lectura de los datos del actor digital en formato .txt, y de esta manera facilitar la comprensión de los datos almacenados y el acceso a las estructuras propias del actor digital (mallas, material, triangulo, vértice y joints).

Se implementan dos funciones: una de escritura y otra de lectura de archivos de texto. La función de escritura se encarga de generar dos archivos con el formato descrito en el Apéndice A. La Tabla 4.1 describe la primer función.

Tabla 4.1 Función de escritura

Función	Entrada	Salida
void archivo()	Se establecen estáticamente los nombres de los archivos datos.txt y datos2.txt	Se obtiene la generación de los dos archivos con sus respectivos contenidos.

La función de lectura “loadModelData” realiza la extracción los datos de los archivos datos.txt y datos2.txt, y carga los datos en las estructuras del actor digital, y se manda a llamar a las funciones siguientes: 1)Agregartexturas, 2) Kinema y 3) Restart. Después del llamado de estas funciones, el actor se encuentre listo para la graficación. La Tabla 4.2 describe la segunda función.

Tabla 4.2 Función de lectura

Función	Entrada	Salida
bool loadModelData(const char *filename, const char *filename2)	Recibe dos punteros con el nombre de los dos archivos datos.txt y datos2.txt	True .- Lectura exitosa. False .- La lectura falló.

4.2. Rutinas de movimiento para las articulaciones

Para generar el movimiento del actor digital se hace uso de la función “Kinema” y “Animation”. La función “Animation” realiza un recorrido de todos los joints del actor digital, esto permite que si hay algún movimiento dentro de algún joint, se multiplican los cambios desde la raíz hasta llegar al joint manipulado y los consecutivos a éste (herencia del nodo padre a los hijos, se aplica la cinemática directa. Además de que el actor digital es un caso especial ya que cada joint es de tipo esfera-y-cavida por lo tanto usa la respectiva matriz de cinemática [1] y [5]).

Para animar los joints del actor digital es necesario definir el nombre del joint y establecer una variable de velocidad por cada eje x, y ó z dependiendo del movimiento requerido, además de definir el lado del cuerpo (izquierdo o derecho); este punto es opcional ya que hay articulaciones en las cuales no es necesario, como el cuello. El nombre del joint, el eje y posiblemente el lado del cuerpo son necesarios para generar un llamado a la función correspondiente que permitirá hacer el cálculo de los grados que moverá la articulación. Por ejemplo, mover el cuello del actor digital hace uso de la función “Movcuello”, ésta permite manipular en los tres ejes al joint llamado “Neck”.

La Figura 7.1 representa el esqueleto y las articulaciones del actor digital y la Tabla 4.3 define las articulaciones y el movimiento con las funciones respectivas y variables que permiten su animación. La descripción del joint puede iniciar con una letra opcional para indicar el lado derecho ‘R’ o izquierdo ‘L’.

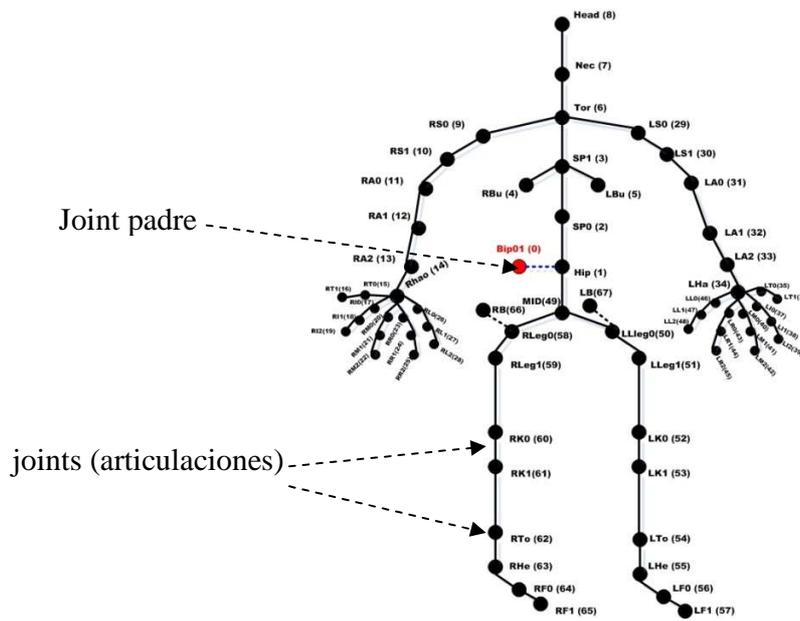


Figura 7.1 Esqueleto del actor digital

Tabla 4.3 Parte 1. Descripción de funciones que permiten manipular los joints del actor digital.

Parte1.

Articulación Humana	Joint del actor digital	Movimiento	Función	Entrada	Salida
Cuello	Nec	Flexión, Hyperextensión Lateral Rotación	Void Movcuello	float rotVec[4] float velocNeck X, float velocNeck Y, float velocNeck Z	float rotVec[4]
Hombro	RS0, LS0	Flexión, Hyperextensión Lateral Rotación	void movhombro	float rotVec[4], float gradoX, float gradoY, float gradoZ, char c	float rotVec[4]

Parte 2

Articulación Humana	Joint del actor digital	Movimiento	Función	Entrada	Salida
Codo	RA1,LA1	Flexión Hyperextensión Supinación Pronación	void MovCodo	float rotVec[4], float grado, char c	float rotVec[4]
Muñeca	RA2,LA2	Flexión Hyperextensión Lateral	void Movmuneca	float rotVec[4], float grado,float grado1, char c,int num	float rotVec[4]
Dedos de mano	RT0,RT1 RI0,RI1 RI2,RI0 RM1,RM2 RR0,RR1R R2,RL0 RL1, RL2 LT0,LT1 LI0,LI1 LI2,LM0 LM1,LM2 LR0,LR1 RL2,LL0 LL1, LL2	Flexión, Hyperextensión Aducción Abducción	void MovdedosMano	float rotVec[4], float gradox, float gradoy, char c)	float rotVec[4]
Cintura	Hip	Flexión, Hyperextensión Lateral Rotación	void MovCintura	float rotVec[4], float gradox,fla t gradoy,fla t gradoz	float rotVec[4]
Ingle	RLeg0,LL eg0	Flexión, Hyperextensión Lateral	void Movingle	float rotVec[4], float grado, float grado1, char c	float rotVec[4]

Parte 3.

Articulación Humana	Joint del actor digital	Movimiento	Función	Entrada	Salida
Rodilla	RK1,LK1	Flexión, Hyperextensión	void movCintura	float rotVec[4], float gradox, float gradoy, float gradoz	float rotVec[4]
Tobillo	RT0,LT0	Flexión, Hyperextensión Lateral	void MovTobillo	float rotVec[4], float grado, float grado1, char c	float rotVec[4]
Dedos de pie	RF0,LF0	Flexión, Hyperextensión	void MovY	float rotVec[4], (float grado)	float rotVec[4]

4.3. Restablecer interfaz *Digital Actor*

La aplicación *Digital Actor* permite la interacción con el usuario mediante dos áreas: graficación y controles. La primera muestra al actor digital y los diversos movimientos que el usuario manipula desde la segunda. Para mejorar el rendimiento de la aplicación, se generan rutinas que actualizan ambas áreas.

El área de graficación restaura al actor digital a su estado inicial por medio del llamado de la función “Restar”, inicializa las variables de las velocidades (las cuales se manipulan los grados que permiten el movimiento para cada articulación), con el propósito de hacer un llamado a la función “Animation”, que se encarga de realizar el recorrido de todas las articulaciones para graficar al actor digital.

El área de controles utiliza la función “inicializa”, la cual realiza un repintado de los controles (barras de desplazamiento, cajas de texto) con valores iniciales. Para que el

usuario pueda hacer uso de las funciones de actualización, se requiere presionar el botón “Restablecer”. La Figura 7.2 representa el uso de este botón.

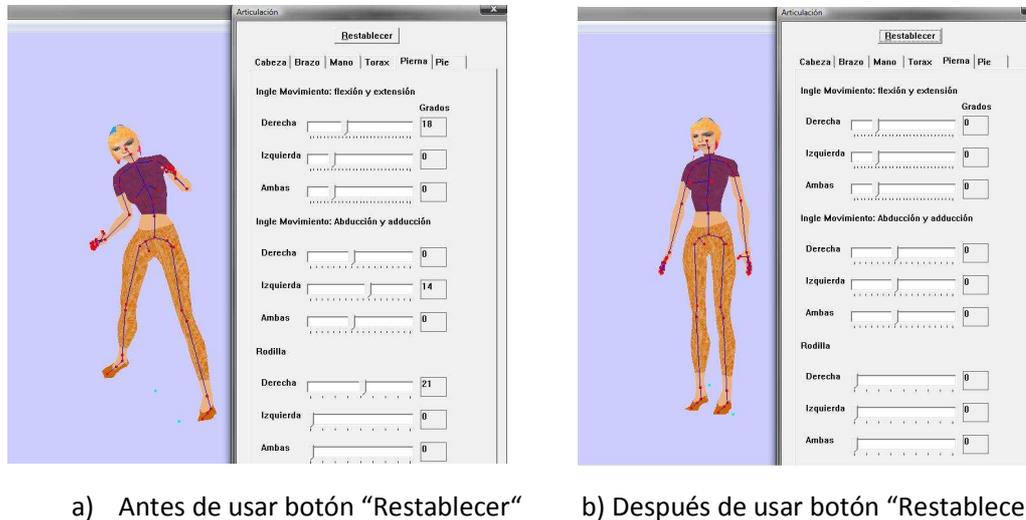


Figura 7.2 Función del botón “Restablecer” de la aplicación *Digital Actor*

4.4. Componentes de la aplicación *Digital Actor*

La interfaz de la aplicación *Digital Actor* facilita la comprensión de los movimientos de las articulaciones humanas, implementa cinemática directa. El actor digital está constituido por cadenas cinemáticas que permiten el movimiento. La interacción usa controles del API de Microsoft Windows (Win 32 API) que están programados en el lenguaje C ++. Esta API requiere la librería <windows.h> para trabajar con ventanas [14].

El desarrollo de aplicaciones con el API de Windows hace uso de dos funciones: la principal denominada “WinMain” y otra de procedimientos “WindowProcedure”. La primer función inicializa la aplicación en tres secciones: 1) *sección de declaración*, consiste en definir las variables que necesita la función, 2) *sección de inicialización*, registra la clase o clases de ventanas, su creación y visualización, además establece el área donde se grafica el actor digital y las rutinas que permiten el control de las articulaciones, y 3) *sección de bucle de mensajes*, la cual define el estado de vida de la aplicación.

La función “WindowProcedure” se encarga del comportamiento de la ventana por medio del procesamiento de la entrada de mensajes los cuales indican alguna función. Por

ejemplo, el mensaje “WM_DESTROY” que se emplea para el cierre de la ventana [6] y [14]. En *Digital Actor* los mensajes que se emplean permiten hacer uso del teclado, crear controles, manejar eventos de los controles y de ventana. La Tabla 4.4 describe las funciones “WinMain” y “WindowProc”.

Tabla 4.4 Definición de las funciones principal y procedimiento.

Funciones de la aplicación	Entrada	Salida
int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nCmdShow)	HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nCmdShow	(int) msg.wParam; Retorno de mensajes para la función de procedimientos “WindowProc”
LRESULT CALLBACK WindowProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lParam)	HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lParam	DefWindowProc(hwnd,uMsg,wParam,lParam) Retorno del procedimiento default.

*La función WindowProc es necesaria para ventanas y cuadros de dialogo [6] y [14].

Para crear controles y ventanas en el API de Windows se usan dos funciones las cuales se diferencia por el tiempo de ejecución [14]. Éstas se describen a continuación:

A) *CreateWindows*.- Esta función crea ventanas, ventanas pop-up y ventanas hijas. Especifica la clase, título, estilo y opcionalmente la posición y tamaño de la ventana. También especifica la ventana padre o propietaria si existe y el menú de la ventana. Además de la creación de ventanas, se pueden generar controles predefinidos por medio de clases que son usados en el parámetro “lpClassName”, especificando cada una de sus características. La tabla 4.5 especifica la función *CreateWindows*.

Tabla 4.5 Definición de la función CreateWindows y las clases de controles utilizadas.

Función	Clase de controles	Controles utilizados en Digital Actor
<pre> HWND CreateWindow(LPCTSTR lpClassName, // puntero al nombre de la clase registrada LPCTSTR lpWindowName, // puntero al nombre de la ventana DWORD dwStyle, // estilo de ventana int x, // posición horizontal de la ventana int y, // posición vertical de la ventana int nWidth, // ancho de la ventana int nHeight, // altura de la ventana HWND hWndParent, // manipulador de la ventana padre o propietaria HMENU hMenu, // manipulador de menú o identificador de ventana hija HANDLE hInstance, // manipulador de la instancia de la aplicación LPVOID lpParam // puntero a los datos para la creación de la ventana); </pre>	<p>BUTTON</p> <p>COMBOBOX</p> <p>DIALOG</p> <p>EDIT</p> <p>LISTBOX</p> <p>MDICLIENT (ventanas cliente)</p> <p>SCROLLBAR</p> <p>STATIC</p> <p>WC_TABCONTROL</p>	<p>BUTTON</p> <p>DIALOG</p> <p>EDIT</p> <p>STATIC</p> <p>WC_TABCONTROL</p>

B) *CreatewindowsEx*.- Es similar a la función CreateWindow, pero las ventanas se crean en tiempo de ejecución [6] y [14]. Tabla 4.6 define a la función *CreatewindowsEx*.

Tabla 4.6 Especificación de función *CreatewindowsEx* y las clases de controles utilizados.

Función	Clase de controles	Controles utilizados en Digital Actor
<pre> HWND CreatewindowEx(DWORD dwExStyle, // estilo extendido de ventana LPCTSTR lpClassName, // puntero al nombre de la clase registrada LPCTSTR lpWindowName, // puntero al nombre de la ventana DWORD dwStyle, // estilo de ventana int x, // posición horizontal de la ventana int y, // posición vertical de la ventana int nWidth, // ancho de la ventana int nHeight, // alto de la ventana HWND hWndParent, // manejador de la ventana padre o propietaria HMENU hMenu, // manejador de menu, o identificador de ventana hija HINSTANCE hInstance, // manejador de la instancia de la aplicación LPVOID lpParam // puntero a los datos de creación de la ventana); </pre>	<p>TRACKBAR_CLASS (Reglas de desplazamiento)</p> <p>DIALOGCLASS (cuadro de diálogos)</p> <p>WNDCLASS (ventanas)</p>	<p>TRACKBAR_CLASS</p> <p>DIALOGCLASS</p> <p>WNDCLASS</p>

Los controles son la forma en que las aplicaciones intercambian datos con el usuario, normalmente se usan dentro de un cuadro de diálogo o cualquier ventana. La Tabla 4.7 describe los controles y su implementación dentro de la interfaz *Digital Actor*.

Tabla 4.7 Descripción de los controles utilizados en *Digital Actor*

Control	Definición	Uso dentro de <i>Digital Actor</i>
Menú	Es una ventana de tipo pop-up, que contiene una lista de comandos u opciones que el usuario puede elegir.	Menú <i>control</i> y <i>ayuda</i> .
Cuadros de Diálogo (Dialog)	Es una ventana de comunicación entre una aplicación Windows y el usuario. Permite agregar controles.	Cuadros de diálogo <i>Articulación</i> , <i>Ayuda</i> y <i>Acerca de Digital Actor</i> .
Botón (Button)	Es un elemento que ejecuta ciertas acciones o para dar funcionalidades a una aplicación.	Botones <i>Restablecer</i> y <i>Aceptar</i> .
Etiquetas (Statics)	Es un control que despliega texto y gráficos.	Se emplean etiquetas para informar al usuario el nombre de <i>articulaciones</i> , <i>grados</i> , <i>ayuda</i> y <i>acerca de Digital Actor</i> .
Cajas de texto (Edit)	Es una ventana de control rectangular que permite al usuario introducir y editar texto desde el teclado.	Representan los grados incrementados o decrementados de las articulaciones.
Reglas de Desplazamiento (Trackbar)	Reglas para la introducción de valores entre rangos definidos.	Se emplean por cada articulación al menos una regla para controlar un movimiento.
Páginas (Tabcontrol)	Es un control que conecta varias ventanas con la correspondiente etiqueta.	Se implementa el control Tab sobre el cuadro de diálogo <i>Articulación</i> con el fin de organizar las diversas partes del cuerpo a través de etiquetas que muestran las articulaciones a manipular.

Un caso particular del control menú para su generación es el uso de la función “CreateMenu” y para agregar elementos por medio de la función “AppendMenu” a diferencia de los anteriores controles que sólo usan funciones CreateWindows y CreateWindowsEx también definidas por el API [6].

4.5. Control de articulaciones

La aplicación *Digital Actor* para poder controlar los rangos de las articulaciones en el área de graficación y de controles, limita los rangos de las diversas articulaciones y dependiendo del movimiento a generar, a través de la función WinMain. Por ejemplo: la articulación muñeca con movimiento de Flexión (movimiento con palma hacia abajo) e Hyperextensión (movimiento con dorso de la mano hacia arriba) permite un limite de rango de 60° en ambos movimientos; en nuestro código se establece un rango de -60° a 60° para lograr el movimiento. De manera similar se trabaja con todas las articulaciones para mantener el control de rango de movimiento.

En el capítulo de resultados se puede observar con mayor detalle los rangos de movimiento para cada articulación del actor digital.

4.6 Interfaz de la aplicación Digital Actor

A continuación se describe la interfaz de *Digital Actor*.

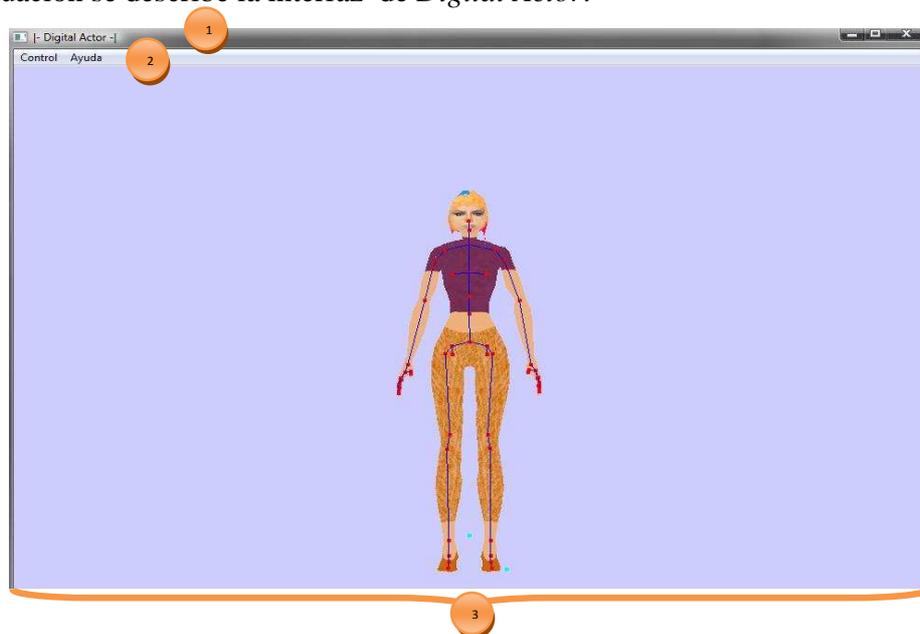


Figura 7.3 Ventana principal *Digital Actor*

Tabla 4.8 Descripción de la Ventana principal *Digital Actor*

Elemento	Nombre	Descripción
1	Barra de título	Nombre de la aplicación <i>Digital Actor</i> .
2	Barra de menú	Opción de elegir el menú Control o Ayuda.
3	Área de graficación	Lugar donde se visualiza al actor digital

Menú Control



Figura 7.4 Ventana principal uso de menú *Control Digital Actor*

Tabla 4.9 Descripción del menú *Control* de *Digital Actor*

Elemento	Nombre	Descripción
1	Opción Articulaciones	Muestra cuadro de diálogo Articulación (permite el control de los joints del actor)
2	Opción Salir	Cerrar <i>Digital Actor</i>

Opción Articulaciones

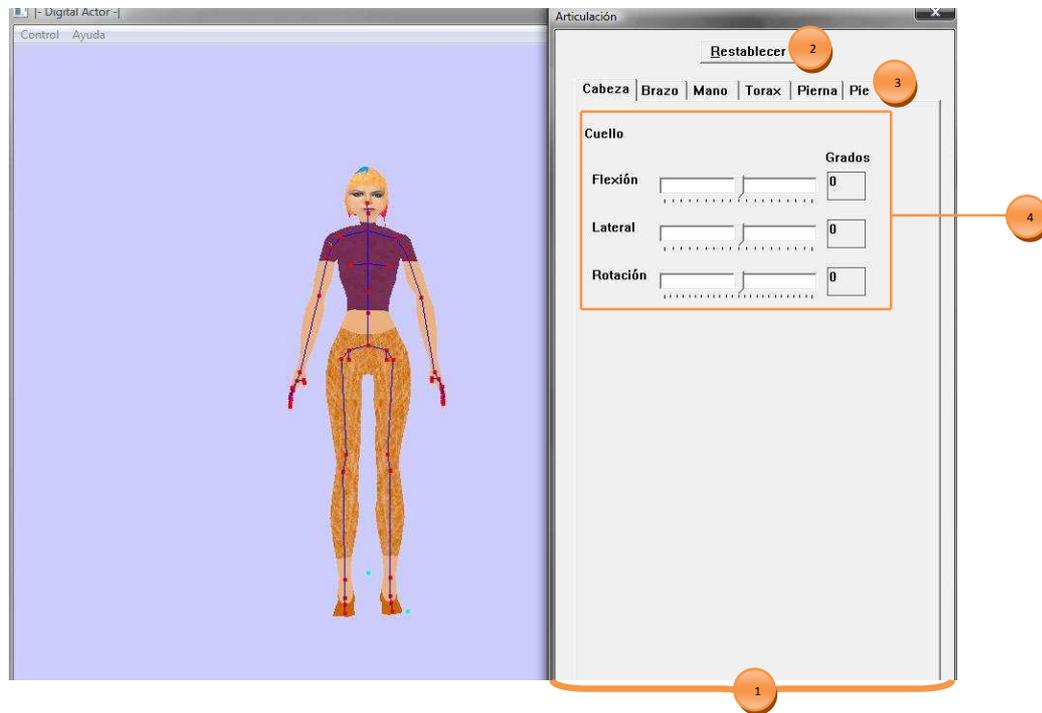


Figura 7.5 Cuadro de diálogo *Articulación* resultado de la opción *Articulaciones*

Tabla 4.10 Descripción del cuadro de diálogo *Articulación* de *Digital Actor*

Elemento	Nombre	Descripción
1	Área de control	Lugar donde usuario manipula componentes para controlar articulaciones del actor digital
2	Botón restablecer	Volver al estado inicial de <i>Digital Actor</i> .
3	Tabs (etiquetas) de las partes del cuerpo humano.	El usuario puede navegar por los diversos Tabs de las partes del cuerpo. El orden va de la cabeza a los pies.
4	Controles de rango	Cada Tab contiene sus propios controles de rango que permiten manipular la articulación respectiva.

*Los controles de rango permite manipular una regla de desplazamiento a la vez para visualizar el incremento y decremento de los grados en el área de graficación.

Menú ayuda



Figura 7.6 Ventana principal uso de menú *Ayuda Digital Actor*

Tabla 4.11 Descripción del menú *Ayuda* de *Digital Actor*

Elemento	Nombre	Descripción
1	Opción <i>Ayuda Digital Actor</i>	Cuadro de Diálogo con instrucciones de uso de la aplicación <i>Digital Actor</i>
2	Opción <i>Acerca de Digital Actor</i>	Cuadro de Diálogo con información sobre la versión de la aplicación.

Opción Ayuda *Digital Actor*.

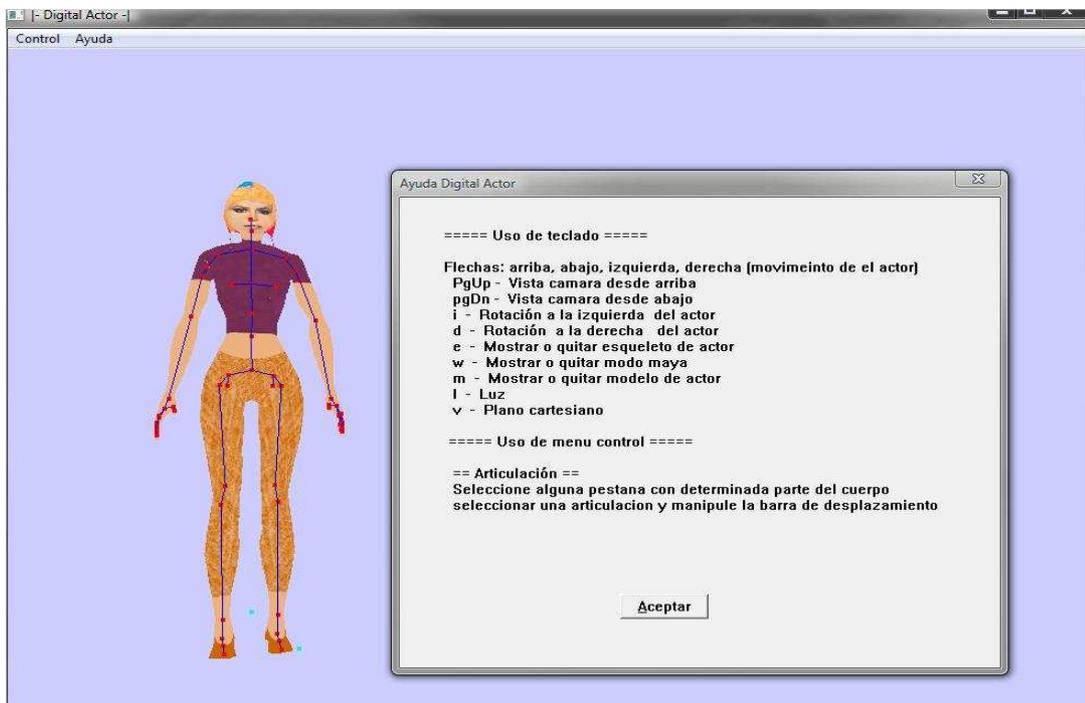


Figura 7.7 Cuadro de diálogo *Ayuda Digital Actor*

Opción Acerca de *Digital Actor*

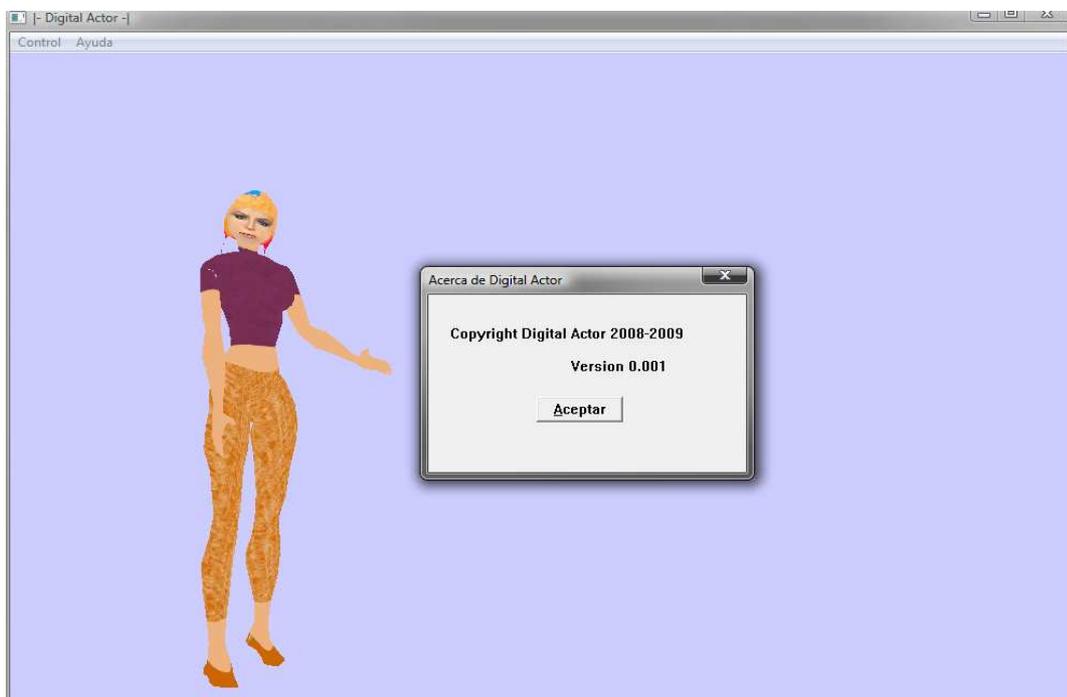


Figura 7.8 Cuadro de diálogo *Acerca de Digital*

Capítulo 5. Resultados de “Digital Actor”

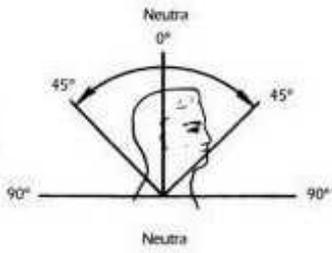
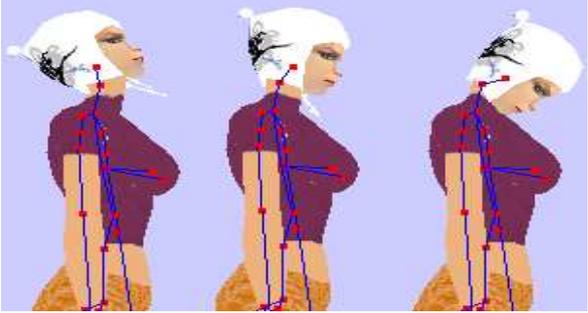
Este capítulo representa algunos movimientos de 8 articulaciones del actor digital. El orden que se sigue es de acuerdo a las características físicas de un humano, iniciando con la cabeza y finalizando con los pies. Cabe mencionar que se logró el movimiento de las 39 articulaciones restantes y que al igual que en los siguientes resultados, se respetan los rangos permitidos.

Todos los movimientos del actor digital tienen un rango establecido de acuerdo a la anatomía de un humano [1] [10] y [15]. Para generar el movimiento, el punto inicial en casi todas las articulaciones parte del grado 0.

5.1 Cabeza

Movimiento flexión y extensión de cuello.- Consiste en que el cuello posee un movimiento hacia delante y hacia atrás en 45°. Para el actor digital se manipula la articulación “Tor” y se genera una animación sobre el eje X teniendo el rango de -45° a los 45°. La Tabla 5.1 representa este resultado.

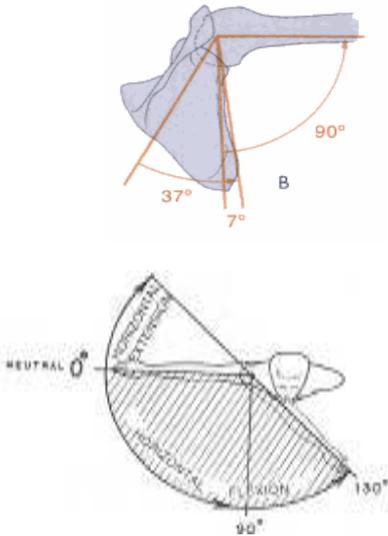
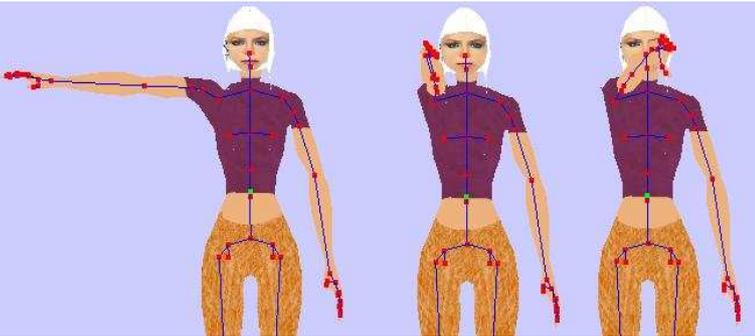
Tabla 5.1 Implementación del movimiento flexión y extensión de cuello

Anatomía	“Digital Actor”
	

5.2 Hombro

Movimiento rotación hacia adentro o afuera del brazo. - El hombro para realizar este movimiento debe situarse en abducción en 90°, esto permite que gire hacia adentro o afuera del brazo en un rango de 0° a 130°. Para el actor digital se manipula la articulación “RS0 (hombro derecho) y LS0 (hombro izquierdo)” y se genera una animación sobre el eje Y a 90°, el eje Z posee el rango de 0° a los 130°, este último eje es el que permite la rotación. La Tabla 5.2 representa este resultado para el hombro derecho.

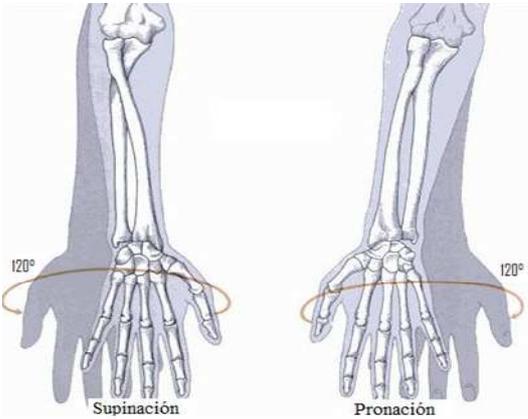
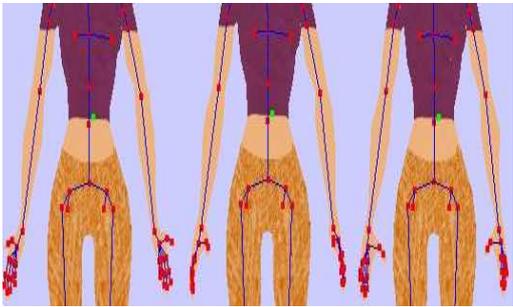
Tabla 5.2 Resultado de implementar el movimiento rotación hacia adentro o afuera del brazo

Anatomía	"Digital Actor"
	

5.3 Brazo

Movimiento pronación y supinación de codo.- Consiste en la rotación del codo de acuerdo a la palma de la mano. El rango permitido es de 0° a 120°. Para el actor digital se manipula la articulación "RA1 (codo derecho) y LA1 (codo izquierdo)" y se genera una animación sobre el eje Z de 80° a -40° para formar los 120°. La Tabla 5.3 muestra el resultado para ambos codos.

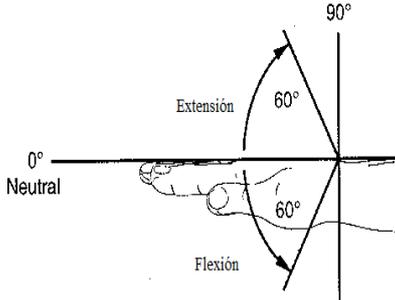
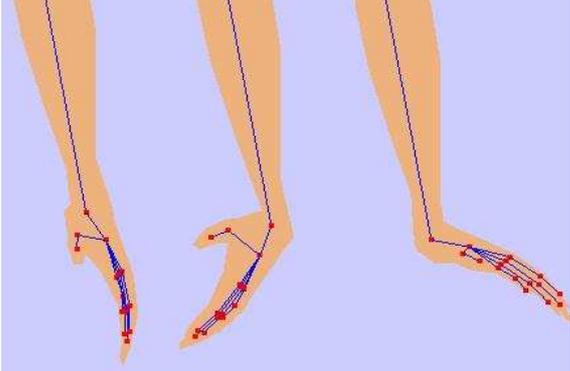
Tabla 5.3 Implementación de movimiento pronación y supinación para ambos codos

Anatomía	"Digital Actor"
	

5.4 Muñeca

Movimiento flexión y extensión de muñecas.- Las muñecas poseen un movimiento de flexión (movimiento con palma hacia abajo) y extensión (movimiento con dorso de la mano hacia arriba) de 60°. Para el actor digital se manipula la articulación “RA2 (muñeca derecha) y LA2 (muñeca izquierda)” y se genera una animación sobre el eje Y de un rango de -60° a 60°. La Tabla 5.4 muestra el resultado para la muñeca izquierda.

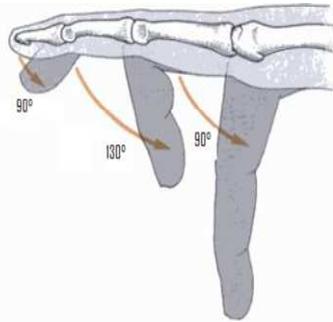
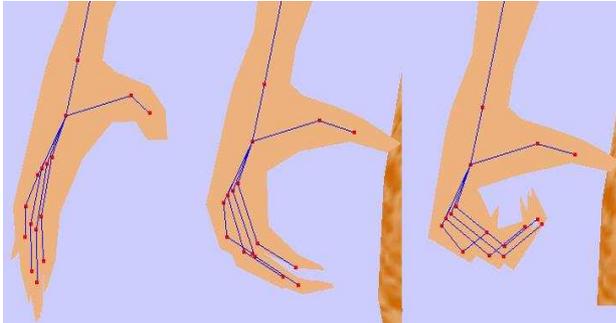
Tabla 5.4 Resultado del movimiento flexión y extensión de muñeca izquierda

Anatomía	“Digital Actor”
	

5.5 Mano

Movimiento flexión y extensión de mano.- Los dedos de la mano poseen movimiento hacia delante y de hacia atrás. El rango de éstos va de 0° a 90° con excepción de una articulación que alcanza los 130° ver Tabla 5.5. Para el Actor digital se manipulan las articulaciones “RT0, RT1, RI0, RI1, RI2, RM0, RM1, RM2, RR0, RR1, RR2, RL0, RL1, RL2 (mano derecha) LT0, LT1, LI0, LI1, LI2, LM0, LM1, LM2, LR0, LR1, LR2, LL0, LL1, LL2 (mano izquierda)” y se genera una animación sobre el eje Y de 0° a 90° o 0° a 130°.

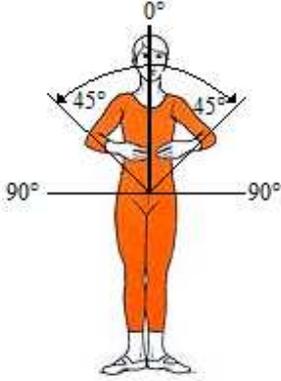
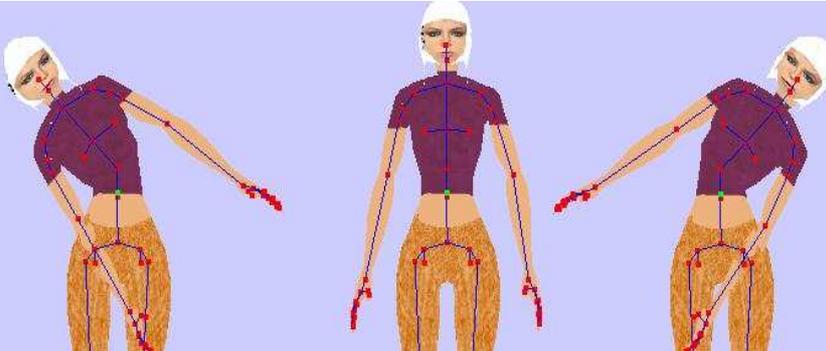
Tabla 5.5 Resultado del movimiento flexión y extensión de la mano

Anatomía	"Digital Actor"
	

5.6 Tórax

Movimiento Abducción y Aducción de cintura.- Consiste en el movimiento lateral izquierdo y derecho de la cintura de 45° . Para el actor digital se manipula la articulación "Hip" y se genera una animación sobre el eje Y con rango de -45° a 45° . La Tabla 5.6 muestra este movimiento.

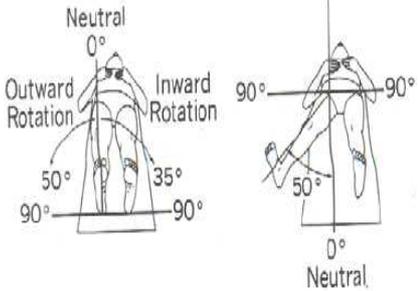
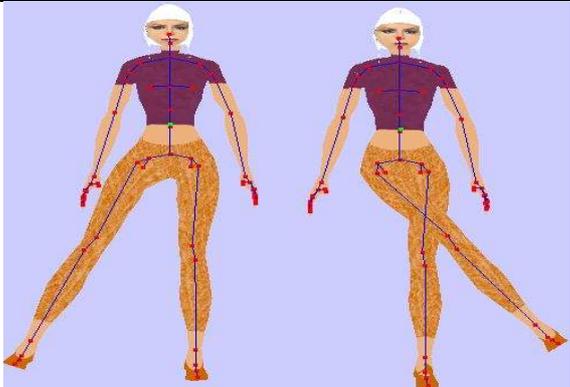
Tabla 5.6 Movimiento abducción y aducción de cintura

Anatomía	"Digital Actor"
	

5.7 Pierna

Movimiento abducción y aducción de ingle.- Las ingles tienen la capacidad de moverse con un rango de movimiento lateral hacia la derecha de 50° y a la izquierda de 35° . Para el actor digital se manipula la articulación "Rleg0 y Lleg0" y se genera una animación sobre el eje Y de -35° a 50° . La Tabla 5.7 muestra el resultado.

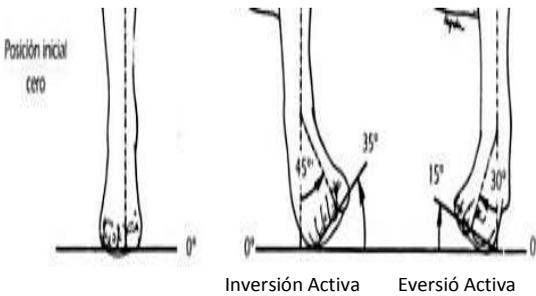
Tabla 5.7 Movimiento abducción y aducción de la ingle izquierda

Anatomía	"Digital Actor"
	

5.8 Pie

Movimiento inversión y eversión activa de tobillo.- Los tobillos del cuerpo tienen una rotación hacia la derecha de 45° o hacia la izquierda de 35°. Para el Actor digital se manipula la articulación "RT0 y LT0" y se genera una animación sobre el eje Z de -35° a 45°. La Tabla 5.8 muestra el resultado para ambos tobillos.

Tabla 5.8 Resultado de movimiento inversión y eversión activa de ambos tobillos

Anatomía	"Digital Actor"
	

6. Conclusiones

Digital Actor es la simulación virtual de un robot físico humano con articulaciones que le permiten realizar movimientos. A través de una interfaz, el usuario puede interactuar con un actor digital.

El proyecto requirió del análisis de la convención y algoritmo de Denavit-Hartenberg, así como de conceptos de unión revolutiva y de esfera y cavidad. Inicialmente, el modelo del actor digital se almacenaba en archivos con extensión .msd3, actualmente se emplean archivos de texto (extensión .txt), lo cual facilita la manipulación de las articulaciones porque se programaron rutinas para cada articulación. Las rutinas implementan conceptos de cinemática directa.

El proyecto consta de 53 articulaciones de las cuales 14 ya fueron trabajadas por [1] y obteniendo un total de 10 DOF, por lo tanto se implementó el movimiento de 39 articulaciones de acuerdo al rango permitido por la anatomía humana obteniendo un total de 70 DOF. Los movimientos implementados se ven afectados por mallas las cuales dan un grado de deformación del movimiento. Por ejemplo, existe una deformación en el movimiento de flexión y extensión de tobillo que da otra interpretación del movimiento. En general, el trabajo realizado en el proyecto refleja el logro de los objetivos propuestos.

Como trabajo futuro se proponen las siguientes líneas de acción: 1) manejo de mallas asociadas a cada joint para afrontar la deformación de las mayas de manera que el movimiento sea lo más cercano al de un humano, 2) realizar animaciones de músculos, ojos y piel, 3) incorporar cinemática inversa al actor digital con el propósito de que se desplace en un mundo virtual. La última línea de acción requerirá de técnicas de planificación de movimiento.

Referencias

- [1] Ademir W. 2006. Cinemática directa aplicada a un actor digital. Universidad Iberoamericana Puebla. Tesis de licenciatura. 2006.
- [2] Arechavaleta G. Esteves C. y Lamound J-P. 2004. Planning fine motions for digital factotum. *Proceedings of the International Conference on Intelligent Robots and Systems (IROS, Septiembre-Octubre, Sendai, Japan)*, 822-827.
- [3] Baerlocher P. Boulic R. 2001. Parametrization and range of motion of the ball-and-socket joint. Publicacion Kluwer Academic.
- [4] Benítez A y Vallejo D. 2007. Diseño y Movimiento para Actores Digitales. <http://www.itpuebla.edu.mx/Eventos/MemoriasyResSemanaInformatica2007/12-%20AntonioBenitezRuiz.pdf>
- [5] Benítez A y Vallejo D. 2007. Forward Kinematics for Virtual Agents. *Engineering Letters*. Vol. 15. Num. 2.10-20.
- [6] Bodnar J. 2007, The Winapi (C Win32 API, No MFC) tutorial. <http://www.zetcode.com/tutorials/winapi/>
- [7] Boulic R. Ulicny B y Thalmann D. 2004. Versatile walk engine. *Journal of Game Development*. Vol. 1. 1-15.
- [8] Casillas S.2000. Robótica: Morfología y cinemática. Centro Universitario de Ciencias Exactas e Ingenierías. Universidad de Guadalajara. México.
- [9] Cuesta B. y Méndez J. 2004. Simulación virtual de robots articulados. Universidad de la Laguna. Departamento de Ingeniería de Sistemas y Automática y arquitectura y Tecnología de Computadoras. España. <http://www.isaatc.ull.es/simrob/doc/proyecto-0.htm>
- [10]Gowitzke A, Milner M. 2000. El cuerpo y sus movimientos: Bases científicas, Editorial Paidotribo. <http://proton.ucting.udg.mx/materias/robotica>

- [11] Jocheim A , Gerke M y Bischoff A. 1999. Modelling and Simulation of Kinematic System. Sitio web de la University of Hagen. Curso de robótica. <http://prt.fernuni-hagen.de/pro/richodl>
- [12] Kwon Y. Ball State University. 2008. Lab#2: Joint Range of Motion. http://www.bsu.edu/web/yknwon/pep294/lab2/rom_lab.html
- [13] Platt Rt. 2007. Introduction to Robotics. <http://www-robotics.cs.umass.edu/~rplatt/>
- [14] Pozo S. y conclase.net. 2001. Usando el API de Windows en C y C++. Win API con Clase, cursos win api32. <http://winapi.conclase.net/curso/index.php>
- [15] Rouvière H, Delmas A, Delmas V. 2005. Anatomía humana: Descriptiva, topográfica y funcional. Editorial El Servier, Edition: 11
- [16] Sánchez C. y Sánchez G. 2000. Manipulación del espacio transformaciones y proyecciones OpenGL. Curso de informática gráfica de la Universidad D Salamanca. Biomedicina, Sistemas Inteligentes y Tecnología Educativa. España. <http://gsii.usal.es/~corchado/igrafica>
- [17] Sánchez O. Robótica: Cinemática de manipuladores. Universidad de Huelva España, Profesores Omar Sánchez, Recursos robotica.http://www.uhu.es/omar_sanchez/indexx.htm
- [18] Sandoval R. 2007. Fundamentos de Robótica. <http://cursos.itchihuahua.edu.mx/course/>
- [19] Xie M. 2003. Fundamentals of Robotics Linking Perception to Action. Editorial World Scientific.

Apéndice

Algoritmo de Denavit-Hartenberg

El algoritmo de Denavit-Hartenberg es una solución para el problema de cinemática directa. Consiste de los siguientes pasos.

- **DH-1.-** Numerar los eslabones comenzando con 1 (primer eslabón móvil de la cadena) y acabando con n (último eslabón móvil). Se numerará como eslabón 0 a la base fija del robot.
- **DH-2.-** Numerar cada articulación comenzando por 1 (correspondiente al primer grado de libertad y acabando en n).
- **DH-3.-** Localizar el eje de cada articulación. Si ésta es rotativa, el eje será su propio eje de giro. Si es prismática, será el eje a lo largo del cual se produce el desplazamiento.
- **DH-4.-** Para i de 0 a $n-1$, situar el eje Z_i , sobre el eje de la articulación $i+1$.
- **DH-5.-** Situar el origen del sistema de la base (S_0) en cualquier punto del eje z_0 . Los ejes x_0 e y_0 se situarán de modo que formen un sistema dextrógiro con z_0 .
- **DH-6.-** Para i de 1 a $n-1$, situar el sistema (S_i) (solidario al eslabón i) en la intersección del eje z_i con la línea normal común a z_{i-1} y z_i . Si ambos ejes se cortasen se situaría (S_i) en el punto de corte. Si fuesen paralelos (S_i) se situaría en la articulación $i+1$.
- **DH-7.-** Situar x_i en la línea normal común a z_{i-1} y z_i .
- **DH-8.-** Situar y_i de modo que forme un sistema dextrógiro con x_i y z_i .

- **DH-9.-** Situar el sistema (S_n) en el extremo del robot de modo que z_n coincida con la dirección de z_{n-1} y x_n sea normal a z_{n-1} y z_n .
- **DH-10.-** Obtener θ el ángulo que hay que girar en torno a z_{i-1} para que x_{i-1} y x_i queden paralelos.
- **DH-11.-** Obtener d_i la distancia medida a lo largo de z_{i-1} , que habría que desplazar (S_{i-1}) para que x_i y x_{i-1} quedasen alineados.
- **DH-12.-** Obtener a_i como la distancia medida a lo largo de x_i (que ahora coincidiría con x_{i-1}) que habría que desplazar el nuevo (S_{i-1}) para que su origen coincida con (S_i).
- **DH-13.-** Obtener α_i como el ángulo que habría que girar en torno a x_i (que ahora coincidiría con x_{i-1}), para que el nuevo (S_{i-1}) coincida totalmente con (S_i).
- **DH-14.-** Obtener las matrices de transformación ${}^{i-1}A_i$.
- **DH-15.-** Obtener la matriz de transformación que relaciona el sistema de la base con el del extremo del robot $T = {}^0A_1, {}^1A_2 \dots {}^{n-1}A_n$.
- **DH-16.-** La matriz T define la orientación (submatriz de rotación) y posición (submatriz de traslación) del extremo referido a la base en función de las n coordenadas articulares.

Especificación de los archivos datos.txt y datos2.txt

Los archivos datos.txt y datos2.txt almacenan la información necesaria para graficar el actor digital. Los datos contenidos en el archivo datos.txt y su estructura es la siguiente:

- El primer renglón contiene el número total de vértices que es igual a 1930. En los siguientes 1930 renglones se define para cada vértice cuatro coordenadas de localización, seguido del identificador del hueso del vértice.
- El renglón 1932 indica el número total de triángulos (2275)
- Del renglón 1933 al 4408 (los siguientes 2275 renglones) contienen 18 datos. Los primeros 9 corresponden a la matriz 3 x 3 de normales (cada fila corresponde a un vector de 3 datos para un vértice del un triángulo), los siguientes 6 datos corresponden a los vectores m_s y m_t de la estructura del actor digital, los últimos 3 datos representan los vértices de un triángulo.

La Figura 8.1 ilustra la representación del formato de los triángulos.

```
-0.115847 10.5009 -0.125749 0 0
2275|
0.0710641 0.0323834 0.996946 }
-0.249862 -0.593507 0.765061 } Matriz 3 x 3 de normales
-0.227109 -0.0504892 0.97256 }
0.234056 0.230156 0.224356 } Datos m_s y m_t
0.8086 0.8184 0.8105 }
0 1 2 | Identificadores de 3 vértices
-0.227109 -0.0504892 0.97256
-0.249862 -0.593507 0.765061
-0.505892 -0.0611139 0.860429
0.224356 0.230156 0.218456
0.8105 0.8184 0.8242
2 1 3
```

Figura 8.1. Representación del formato de los triángulos.

El archivo datos2.txt y su estructura es la siguiente:

- El primer renglón indica el número total de mallas (grupos) que es igual a 2, las cuales se describen por bloque
- El primer renglón de un bloque de malla tiene el índice del material de la malla, el segundo renglón contiene el número total de triángulos que usa la malla. Posteriormente, de acuerdo a este número, se encuentran almacenado el índice de cada triángulo

- Después de describir las mallas, se almacenan los datos de los materiales. Se define un renglón con el número total de materiales el cual tiene el valor de 1, en los siguientes renglones, se hallan 4 datos que corresponden a la luz ambiental (vector de luz ambiental de 3 elementos), 4 datos que corresponden a la luz difusa (vector de luz difusa de 3 elementos), 4 datos que corresponden a la luz especular (vector de luz especular de 3 elementos), 4 datos que corresponden a la luz emisiva (vector de luz emisiva de 3 elementos), un solo dato de brillo y por último, una cadena de caracteres que es el nombre de la textura o material, por ejemplo, cuerpo.bmp.

Después de los materiales, se localizan los datos de los joints (articulaciones), en un renglón está almacenado el número total de joints que es igual a 68. Para cada uno de los 68 joints se almacena el nombre del padre del joint, 3 coordenadas de rotación, 3 coordenadas de traslación, el nombre del joint y el número de joint padre. Existe una excepción en el primer joint almacenado, ya que éste es el joint padre el cual tiene asignado '#' como nombre del nodo padre y número -1. La Figura 8.2 ilustra la representación del material y joint padre.

```

<< / *
1
0.2 0.2 0.2 1
0.8 0.8 0.8 1
0 0 0 1
0 0 0 1
0
cuerpo.bmp
68
# 3.09651 0 0 -0.001634 7.59853 0.800002 Bip01 -1 → Joint Padre
Bip01 0 0 0 0 0 0.71965 Hip 0
Hip 1.43735 0.003591 0 0 -0.50812 -0.131141 Sp0 1
Sp0 0 0 0 0 0 0.730531 Sp1 2
Sp1 0 0 0 -0.430518 -0.592267 -0.227033 RBu 3

```

Figura 8.2. Representación del formato del material y joint padre.

Tabla 6.1 Grados de libertad de articulaciones de *Digital Actor*

Articulación	Joint	Numero de DOF por articulación	Totales de DOF
Cuello	Nec	3	3
Hombros	RS0, LS0	3	6
Codos	RA1, LA1	2	4
Muñecas	RA2, LA2	2	4
Dedos mano	RT0, RT1, RI0, RI1, RI2, RM0, RM1, RM2, RR0, RR1, RR2, RL0, RL1, RL2 LT0, LT1, LI0, LI1, LI2, LM0, LM1, LM2, LR0, LR1 RL2, LL0, LL1, LL2	2	38
Cintura	Hip	3	3
Ingle	RLeg0, LLeg0	2	4
Rodillas	RK1, LK1	1	2
Tobillo	RT0, LT0	2	4
Dedos pie	RF0, LF0	1	2

Total=70

Degree of Freedom (DOF).- Grados de libertad