# **INSTITUTO TECNOLÓGICO DE CHIHUAHUA** DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

# "ANIMACIÓN DE MARIONETAS DIGITALES POR FUSIÓN DE SENSORES RGBD"

## **TESIS**

QUE PARA OBTENER EL GRADO DE

# MAESTRA EN CIENCIAS EN INGENIERÍA ELECTRÓNICA

**PRESENTA:** 

# ING. ELISABET GONZÁLEZ JUÁREZ

DIRECTOR DE LA TESIS: *M.C. ALBERTO PACHECO GONZÁLEZ* 



CHIHUAHUA, CHIH., JUNIO 2017

Dr. Enrique Cabrero Mendoza. Director de CONACYT.

> At´n. Mtro. Pablo Rojo Calzada Director de Becas.

Presente.

Por este conducto aprovecho la ocasión para saludarlo e informarle que a la fecha he obtenido el Grado de Maestría en Ciencias de Ingeniería Electrónica en la División de Estudios de Posgrado e Investigación del Instituto Tecnológico de Chihuahua. Motivo por el cual agradezco todo el apoyo brindado por esa institución que usted representa. El otorgamiento de esta beca me permitió dedicarme de tiempo completo a la realización de mis estudios de posgrado y de esta manera lograr el cumplimiento del objetivo principal del convenio establecido.

Sin otro particular por el momento, me es grato quedar de usted como su segura servidora, no sin antes reiterar mi agradecimiento. Muchas gracias.

Atentamente

*Ing. Elisabet González Juárez* Exbecaria CONACYT No. 570201

c.c.p M.F. Luis Cardona Chacón. Jefe de la División de Posgrado e Investigación

"La enseñanza que deja huella no es la que se hace de cabeza a cabeza, sino de corazón a corazón".

Howard G. Hendricks

*"Más vale adquirir sabiduría que oro; más vale adquirir inteligencia que plata".* 

Proverbios 16:16 (NVI)

## Dedicatoria

A Dios, el creador todopoderoso. A Jesús, el autor y consumador de la fe. A Él, quien dio todo por mí y me otorgó su invaluable amor y perdón.

A mi familia, por haberme apoyado en todo momento, por creer en mí y por hacer lo que fuese necesario para que pudiera culminar esta etapa de mi vida.

## Agradecimientos

Agradezco primeramente a Dios por haberme permitido vivir esta etapa, por darme la oportunidad de terminar esta tesis y por darme las herramientas necesarias para lograrlo. Gracias a Él pude obtener un logro más en mi vida.

Agradezco también a mis padres Armando y Silvia por siempre estar ahí, por darme su cariño y amor. Gracias por cada una de sus enseñanzas y consejos, gracias porque siempre buscaron la manera de ayudarme a salir adelante.

A mi hermano David, gracias por todo tu apoyo. Aunque te encontrabas lejos siempre estabas al pendiente de mí, preocupándote por mis estudios.

Agradezco a mi director de tesis, M.C. Alberto Pacheco González, por haberme aceptado como tesista. Gracias por su paciencia y apoyo, y por todas las cosas que me enseñó, de verdad aprendí mucho durante mi estancia en el laboratorio.

A mis sinodales, Dr. Isidro Robledo, Dr. Juan Ramírez y Dr. Javier Vega. Muchas gracias por cada una de las aportaciones que hicieron a mi trabajo, fue gracias a esas observaciones que pude mejorar muchas cosas de mi tesis.

Al Dr. René Cruz, gracias por estar al tanto de mi trabajo y por estar presente en cada una de las revisiones de avances de tesis, sus comentarios me ayudaron a avanzar con mi trabajo.

A cada uno de mis profesores: M.C. Enrique Hernández Vega, M.C. Jesús Robles, Dr. Pedro Acosta, Dr. Javier Vega, M.C. Alberto Pacheco, Dr. Isidro Robledo, Dr. Pedro Márquez, Dra. Didia Salas y Dr. Francisco Corral, y al M.C. Edgar Trujillo y el M.C. Jorge Aguirre. Gracias por impartirme un poco sus conocimientos, con ellos pude concluir mis estudios de maestría de una mejor manera. Agradecimientos especiales a la Dra. Didia Salas, solo fue una materia la que me impartió pero sus enseñanzas fueron más allá de eso. Gracias por siempre recibirme y ayudarme a resolver todas mis dudas.

También agradezco a la Dra. Vania Álvarez y a la maestra Blanca Valdez por su participación en las pruebas piloto. De igual forma agradezco a sus alumnos por ayudarnos en dichas pruebas y en general a todas las personas que colaboraron en las pruebas piloto.

A todo el personal administrativo y no administrativo de la DEPI que me ayudó en muchos aspectos y áreas de mi vida durante mis estudios de maestría, especialmente a la coordinadora M.C. Alma Corral por todas sus atenciones y sobre todo por su paciencia.

A mis compañeros de generación: Carlos, Esteban, Víctor, Edgar, Nadia, Carreón, Luis, Chava, Hery, Sosa y Rubén, gracias por su compañía, por todos esos momentos de convivencia y por su amistad, muchas gracias. Les deseo mucho éxito en todo lo que hagan.

Al resto de mis compañeros de la maestría: Misa, David, Meny, Cañedo, Alan, Piña, Arnoldo, Luis, Abimael, Miguel y los demás del laboratorio de visión y otros laboratorios, gracias por permitirme conocerlos. Cañedo, gracias por tu ayuda con la tarjeta Raspberry Pi.

A mis compañeros de laboratorio: Gaby, maestra Angélica y Aaron, y a mis compañeros del Tec 2 de la materia de Dispositivos Móviles. Fue muy grato conocerles, me la pasé muy bien trabajando con ustedes.

Un agradecimiento especial a todos mis hermanos del Centro Cristiano Vida Nueva, en particular al grupo de jóvenes y adolescentes ALIVE. Fueron años muy pesados para mí, pero gracias a ustedes pude disfrutar de muchos momentos alegres que me motivaron a seguir adelante. Al resto de mis amigos y familia, muchas gracias por alentarme a seguir adelante. Gracias por cada consejo, llamada y mensaje, realmente fueron una gran motivación para esforzarme cada vez más. Jusil, muchas gracias por tu amistad, gracias por escucharme y ayudarme con las cosas de la escuela.

Rubén, te agradezco tanto por tu apoyo, por ser no solo mi compañero sino mi mejor amigo. Gracias por estar ahí en los mejores y peores momentos y por ayudarme a no rendirme nunca. Gracias por tu compañía, por todas esas pláticas y momentos juntos, por explicarme las cosas que no entendía, por todo, muchas gracias. Te deseo mucho éxito en todo lo que hagas y que seas feliz siempre.

A todos los que de alguna manera me apoyaron: ¡Muchas gracias!

#### RESUMEN

## ANIMACIÓN DE MARIONETAS DIGITALES POR FUSIÓN DE SENSORES RGBD

Elisabet González Juárez Maestra en Ciencias en Ingeniería Electrónica División de Estudios de Posgrado e Investigación del Instituto Tecnológico de Chihuahua Chihuahua, Chih. 2017 Director de Tesis: M.C. Alberto Pacheco González

Las interfaces naturales de usuario permiten la interacción entre el usuario y la aplicación directamente a través del reconocimiento de voz, gestos, posturas, retroalimentación táctil, ocular o corporal. Algunas tecnologías emergentes de visión por computadora, entre las cuales destacan los sensores de profundidad RGBD, similares al sensor Kinect<sup>®</sup>, han posibilitado el desarrollo de aplicaciones en tiempo real a un bajo costo para manipular directamente marionetas digitales capturando el movimiento del usuario ubicado frente a un sensor RGBD. Un sistema de este tipo puede ser de utilidad, como lo es en este caso, para representar personajes animados dentro de una narrativa digital gracias a la animación resultante de las marionetas digitales.

En la presente tesis, se investigaron y analizaron diversos trabajos relacionados con la animación basada en marionetas digitales utilizando captura de movimiento vía sensores RGBD. En dicha revisión de literatura se detectó que la gran mayoría de los trabajos solo utilizan un sensor para trabajar con la aplicación. Debido a lo anterior, surgió el interés de desarrollar este trabajo de tesis, cuyo objetivo principal consiste en diseñar un sistema capaz de trabajar con múltiples sensores RGBD para animar marionetas digitales en tiempo real.

Para lograr dicho objetivo fue necesario diseñar o simular clientes de captura de movimiento para cada sensor RGBD involucrado. Se realizó el diseño de un protocolo de tiempo real, el cual fue implementado en WebRTC y Socket.IO, para enviar a un servidor web dedicado, todos los datos provenientes de cada módulo de captura. Dicho servidor dedicado se utilizó como unidad central de procesamiento para hacer la fusión de datos de los sensores RGBD y obtener como

resultado la animación de una marioneta digital. En la validación del sistema, se simularon tanto los clientes de captura como la animación de la marioneta digital. Para la simulación se usaron datos de tipo texto, obteniendo como resultado un correcto envío y recepción de datos en tiempo real.

A través de la prueba del sistema mediante simulaciones, fue posible validar la plataforma FSA-MD y el protocolo implementado para animar marionetas digitales en tiempo real, empleando múltiples sensores RGBD, mediante un prototipo limitado al uso de datos convertidos a textos. Este trabajo por otra parte, también ha contribuido a demostrar que la plataforma FSA-MD no está limitada a ser usada solamente en la animación de marionetas digitales, sino que puede ser útil para fusionar en tiempo real diversos flujos de datos multimedia provenientes de diversas fuentes y para muy diversas aplicaciones.

## CONTENIDO

LIST	A DE	E FIGURAS	xvi
LIST	A DE	TABLAS	xxii
I. INT	RO	DUCCIÓN	1
II. AN	ITEC	EDENTES	4
2.1	Pro	oducción de narrativas digitales	4
2.2	Ela	boración de marionetas digitales	8
2.3	Τé	cnicas para la animación de marionetas digitales	11
2.	3.1	Animación mediante la técnica stop motion	12
2.	3.2	Animación 2D	13
2.	3.3	Animación mediante la captura de movimiento	13
2.	3.4	Animación 3D	15
2.4	Tra	ansmisión de flujos de datos multimedia en tiempo real	16
2.5	Re	des de tiempo real de sensores RGBD	17
III. M	ARC	O TEÓRICO	19
3.1	Na	rratología	19
3.	1.1	Narrativa	19
3.	1.2	Narrativa textual	20
3.	1.3	Historia	20
3.	1.4	Fábula	20
3.2	Na	rrativas digitales	21
3.3	Ма	rionetas digitales	21
3.4	Gra	áficos por computadora	22
3.	4.1	Gráficos de mapas de bits	22
3.	4.2	Gráficos vectoriales	24
3.	4.3	Sistema de coordenadas de pixeles	25
3.	4.4	Transformaciones 2D	26
3.5	SV	G	27
3.	5.1	Estructura de un documento SVG	27

3.	5.2	Formas básicas	28
3.6	Inte	eracción hombre-máquina	30
3.7	Inte	eracción multimodal	30
3.8	Inte	erfaces de usuario	31
3.9	Inte	erfaces naturales de usuario	32
3.9	9.1	Interfaces de usuario tangibles	32
3.	9.2	Interfaces de usuario de audio	33
3.	9.3	Interfaces de usuario de video	33
3.10	Ca	ptura de movimiento	33
3.1	0.1	Sistemas magnéticos	34
3.1	0.2	Sistemas electromecánicos	34
3.1	0.3	Sistemas ópticos	34
3.1	0.4	Sensores RGBD	35
3.11	Inte	erfaces de programación de aplicaciones	36
3.12	Apl	licaciones multimedia	37
IV. PF	ROT	OCOLOS DE RED DE TIEMPO REAL	38
4.1	Arc	uitecturas para la comunicación de redes	39
4.	1.1	Modelo OSI	39
4.	1.2	Modelo TCP/IP	42
4.	1.3	Modelo para loT	45
4.2	Sei	rvicios web	47
4.2	2.1	Secuencia de preservación	47
4.2	2.2	Sincronización de unidades de datos	47
4.2	2.3	Control de flujo	48
4.2	2.4	Libre de error	48
4.2	2.5	Reinicio del servicio	49
4.2	2.6	Establecimiento y liberación de la conexión	49
4.2	2.7	Cambio de modo	50
4.3	Pro	otocolos de comunicación	53

4.	3.1	IP	53
4.	3.2	ТСР	54
4.	3.3	UDP	54
4.	3.4	RTP	54
4.4	We	bRTC	55
4.	4.1	Ventajas del uso de WebRTC	56
4.	4.2	Arquitectura web	56
4.	4.3	Función RTC en el navegador	57
4.	4.4	Escenario de conferencias WebRTC	57
4.	4.5	Proceso de comunicación WebRTC	58
4.	4.6	Arquitectura WebRTC	59
4.5	Pru	ebas de software	61
V. PF	ROTO	DCOLO DE TIEMPO REAL PARA LA FUSIÓN DE SENSORES	64
RGBI	D		
5.1	Pla	taforma FSA-MD	64
5.2	Mó	dulos de captura	65
5.	2.1	Etapa de adquisición	65
5.	2.2	Etapa de procesamiento	75
5.	2.3	Etapa de visualización	79
5.3	Rep	productores de animación de marionetas	81
5.4	Ser	vidor para la fusión de datos de sensores RGBD	84
5.5	Pro	tocolo de tiempo real para la fusión de datos multimedia	91
VI. RI	ESU	LTADOS Y CONCLUSIONES	100
6.1	Pro	ducción de marionetas digitales	100
6.2	Pro	ducción de narrativas digitales	106
6.3	Dis	eño e implementación de los módulos de captura	109
6.4	Imp	lementación del servidor web	111
6.5	Imp	elementación del protocolo de tiempo real	112
6.6	Ani	mación de marionetas digitales mediante la plataforma FSA-MD	114

REFERENCIAS	116
ANEXO 1	134
APÉNDICE A	135
APÉNDICE B	177
APÉNDICE C	226
APÉNDICE D	253

## LISTA DE FIGURAS

## **II. ANTECEDENTES**

Figura 2.1.	Sitio web desarrollado por W. Tsou <i>et al</i> . [7].	4
Figura 2.2.	INSCAPE versión beta [8].	5
Figura 2.3.	Editor de escenarios para juegos de aventura [9].	6
Figura 2.4.	Página de inicio de MoViE en un celular Nokia N95 [15].	7
Figura 2.5.	Aplicación móvil Episodium [31].	8
Figura 2.6.	Personaje trazado en Adobe Illustrator <sup>®</sup> .	9
Figura 2.7.	Dibujo hecho a lápiz y papel: a) foto real del dibujo, b) dibujo digitalizado.	10
Figura 2.8.	Dibujo 2D visto desde diferentes perspectivas [40].	10
Figura 2.9.	Clasificación general de las técnicas de animación.	11
Figura 2.10.	Clasificación de las técnicas de animación tradicional.	12
Figura 2.11.	Clasificación de las técnicas de animación digital.	13
Figura 2.12.	Arquitectura inicial del sistema de animación de marionetas en	15
	tiempo real para un solo sensor RGBD [6], [73].	
Figura 2.13.	Escenario del gimnasio en línea propuesto por F. Cassola et al.	18
	[95].	
III. MARCO 1	TEÓRICO	
Figura 3.1.	Representación de los niveles de análisis de una narrativa.	20
Figura 3.2.	Arreglo de pixeles.	22
Figura 3.3.	Ejemplo de una imagen en escala de grises: a) imagen original,	23
	b) imagen en escala de grises, c) niveles de gris de una porción	
	de la imagen.	
Figura 3.4.	Componentes RGB de una imagen: a) imagen original, b)	23
	componente rojo, c) componente verde, d) componente azul.	
Figura 3.5.	Gráfico de mapa de bits escalado: a) imagen original, b) imagen	24
	escalada.	

Comparación entre gráficos: a) gráfico de mapa de bits, b)	25
gráfico de mapa de bits escalado, c) gráfico vectorizado.	
Sistema de coordenadas cartesianas bidimensionales.	25
Sistema de coordenadas de pixeles.	26
Diferentes tipos de transformaciones 2D: a) gráfico original, b)	26
reflexión, c) rotación, d) inclinación, e) escala, f) traslación.	
Estructura básica de un documento SVG.	27
Flujo de información en HCI.	30
Clasificación de las interfaces de usuario.	31
Clasificación de las NUI de acuerdo a los 5 sentidos del ser	32
humano.	
Mapa de profundidad de un sensor RGBD.	35
Campo de visión de un sensor RGBD.	36
Clasificación de las aplicaciones multimedia.	37
OLOS DE RED DE TIEMPO REAL	
Intercambio de mensajes en un protocolo simple.	38
Modelo OSI de 7 capas.	39
Arquitectura TCP/IP de 4 capas.	42
Arquitectura básica de 3 capas para IoT.	45
Arquitectura SoA genérica.	46
Arquitectura IoT-A.	46
Servicio orientado a bloques (imagen superior), servicio	48
orientado a transmisión (imagen inferior).	
Fases de operación de una conexión.	49
Modo de operación <i>simplex</i> .	51
Flujo de información de una estación de radio a un radio.	51
Modo de operación half-duplex.	52
Flujo de información entre dos radios portátiles (walkie-talkie).	52
Modo de operación full-duplex.	53
	Comparación entre gráficos: a) gráfico de mapa de bits, b) gráfico de mapa de bits escalado, c) gráfico vectorizado. Sistema de coordenadas cartesianas bidimensionales. Sistema de coordenadas de pixeles. Diferentes tipos de transformaciones 2D: a) gráfico original, b) reflexión, c) rotación, d) inclinación, e) escala, f) traslación. Estructura básica de un documento SVG. Flujo de información en HCI. Clasificación de las interfaces de usuario. Clasificación de las NUI de acuerdo a los 5 sentidos del ser humano. Mapa de profundidad de un sensor RGBD. Campo de visión de un sensor RGBD. Clasificación de las aplicaciones multimedia. <b>DOS DE RED DE TIEMPO REAL</b> Intercambio de mensajes en un protocolo simple. Modelo OSI de 7 capas. Arquitectura TCP/IP de 4 capas. Arquitectura básica de 3 capas para IoT. Arquitectura SoA genérica. Arquitectura loT-A. Servicio orientado a bloques (imagen superior), servicio orientado a transmisión (imagen inferior). Fases de operación de una conexión. Modo de operación <i>simplex.</i> Flujo de información de una estación de radio a un radio. Modo de operación <i>half-duplex.</i> Flujo de información entre dos radios portátiles ( <i>walkie-talkie</i> ). Modo de operación <i>full-duplex.</i>

Figura 4.14.	Flujo de información en la comunicación vía teléfono.	53
Figura 4.15.	Modelo de un navegador web.	56
Figura 4.16.	Función RTC en el navegador web.	57
Figura 4.17.	Topología de red tipo malla.	58
Figura 4.18.	Topología de red tipo estrella.	58
Figura 4.19.	Proceso general de comunicación WebRTC.	58
Figura 4.20.	Arquitectura WebRTC.	60
Figura 4.21.	Triángulo WebRTC.	60
Figura 4.22.	Trapezoide WebRTC.	61
Figura 4.23.	Modelo de calidad interna y externa según el ISO 9126 [187].	62
Figura 4.24.	Principales tipos de prueba.	63
V. PROTOCO	OLO DE TIEMPO REAL PARA LA FUSIÓN DE SENSORES	
RGBD		
Figura 5.1.	Sensor PrimeSense <sup>®</sup> Carmine 1.08.	64
Figura 5.2.	Plataforma FSA-MD.	65
Figura 5.3.	Sensor Intel <sup>®</sup> RealSense diseñado por Creative Technology <sup>®</sup> ,	66
	modelo VF0800.	
Figura 5.4.	Programa de ejemplo Sample-NiSimpleViewer para el sensor	67
	PrimeSense <sup>®</sup> : a) visualización de una escena, b) visualización	
	de una persona en la escena.	
Figura 5.5.	Programa de ejemplo Sample-NiUserTracker para el sensor	68
	PrimeSense <sup>®</sup> .	
Figura 5.6.	Esqueleto de 15 puntos proporcionado por OpenNI® para el	69
	sensor PrimeSense <sup>®</sup> .	
Figura 5.7.	Esqueleto de 10 puntos utilizado para la plataforma FSA-MD.	69
Figura 5.8.	Múltiples usuarios identificados por el sensor PrimeSense®.	70
Figura 5.9.	Opción para importar librerías en Processing 2.	71
Figura 5.10.	Administrador de librerías de Processing 2.	71
Figura 5.11.	Conexión del sensor RGBD a la computadora.	72

Proceso de adquisición de datos del software de captura.	72
Módulo de captura: mapa de profundidad del sensor RGBD.	73
Módulo de captura: reconocimiento de un usuario.	73
Módulo de captura: pantalla de inicio.	74
Módulo de captura: movimientos de una persona representados	74
en una marioneta digital.	
Ejemplo de un <i>path</i> en un gráfico SVG.	75
Ejemplo de una agrupación en un gráfico SVG.	76
Ubicación de pivotes en una marioneta digital.	76
Representación en código SVG de un grupo.	77
Representación sintética y simbólica en Java del mapeo de	78
pivotes.	
Módulo de captura: caso de visualización con dos usuarios	80
situados frente al sensor RGBD.	
Módulo de captura: pérdida del reconocimiento del usuario	80
principal.	
Módulo de captura: pérdida de la imagen por falta del	81
reconocimiento del usuario.	
Funciones del reproductor de animación de marionetas: a)	82
sistema inicial, b) zoom out, c) rotación a la derecha.	
Escenario de dos personas para el uso del reproductor de	82
animación de marionetas.	
Escenario de una persona para el uso del reproductor de	83
animación de marionetas: a) control de funciones de animación,	
b) realización de movimientos.	
Tarjeta Raspberry Pi 2 modelo B.	84
Escritorio del sistema operativo Raspbian con Pixel.	85
Archivo package.json.	86
	Proceso de adquisición de datos del software de captura. Módulo de captura: mapa de profundidad del sensor RGBD. Módulo de captura: reconocimiento de un usuario. Módulo de captura: movimientos de una persona representados en una marioneta digital. Ejemplo de un <i>path</i> en un gráfico SVG. Ubicación de pivotes en una marioneta digital. Representación en código SVG de un grupo. Representación sintética y simbólica en Java del mapeo de pivotes. Módulo de captura: caso de visualización con dos usuarios situados frente al sensor RGBD. Módulo de captura: pérdida del reconocimiento del usuario principal. Módulo de captura: pérdida de la imagen por falta del reconocimiento del usuario. Funciones del reproductor de animación de marionetas: a) sistema inicial, b) <i>zoom out</i> , c) rotación a la derecha. Escenario de dos personas para el uso del reproductor de animación de marionetas: a) control de funciones de animación, b) realización de movimientos. Tarjeta Raspberry Pi 2 modelo B. Escritorio del sistema operativo Raspbian con Pixel. Archivo <i>package.json</i> .

Figura 5.31.	Página de inicio de la aplicación web para la plataforma FSA-	87
	MD.	
Figura 5.32.	Pantalla de acceso a los clientes de captura.	87
Figura 5.33.	Pantalla de acceso a los clientes de visualización.	88
Figura 5.34.	Formulario de acceso para los clientes de captura.	88
Figura 5.35.	Mensaje de error: no se ha proporcionado un nombre de	89
	usuario.	
Figura 5.36.	Sala de producción de animación de marionetas digitales.	90
Figura 5.37.	Lista de usuarios conectados.	90
Figura 5.38.	Consola del servidor web.	91
Figura 5.39.	Representación del protocolo de red de tiempo real basado en	92
	una topología de red tipo estrella.	
Figura 5.40.	Comunicación cliente-servidor con Socket.IO.	92
Figura 5.41.	Diagrama de secuencia de intercambio de mensajes entre dos	93
	clientes web con Socket.IO.	
Figura 5.42.	Diagrama de secuencia de conexión de múltiples clientes a una	94
	sala.	
Figura 5.43.	Área para datos de entrada.	95
Figura 5.44.	Vista previa de la animación.	95
Figura 5.45.	Recepción de datos de animación en el servidor web.	95
Figura 5.46.	Transmisión punto a punto ( <i>unicast)</i> con Socket.IO.	96
Figura 5.47.	Transmisión punto a multipunto (broadcast) con Socket.IO.	97
Figura 5.48.	Transmisión punto a multipunto (multicast) con Socket.IO.	98
Figura 5.49.	Esquema de validación del protocolo de red de tiempo real	99
	desarrollado.	
Figura 5.50.	Cliente de visualización.	99
VI. RESULT	ADOS Y CONCLUSIONES	
Figura 6.1.	Etapas de desarrollo del trabajo de tesis.	100

- Figura 6.2. Voluntarios para las pruebas de validación del procedimiento 103 MD-01.
- Figura 6.3. Voluntarios para la pruebas de validación del procedimiento 104 MD-02.
- Figura 6.4. Gráfico comparativo del tiempo utilizado por cada método para 104 diseñar una marioneta digital.
- Figura 6.5. Estudiantes durante las pruebas piloto de mayo de 2015. 106
- Figura 6.6. Estudiantes durante las pruebas piloto de noviembre de 2015. 108
- Figura 6.7. Marioneta rediseñada: a) marioneta original, b) marioneta con 109 objeto integrado.

## APÉNDICE D

Figura D1.Cantidad de muestras de acuerdo al método utilizado.254Figura D2.Cantidad de muestras de acuerdo al grupo de edad.254

## LISTA DE TABLAS

III. MARCO	TEÓRICO	
Tabla 3.1.	Formas básicas de SVG.	28
IV. PROTOC	COLOS DE RED DE TIEMPO REAL	
Tabla 4.1.	Protocolos para la capa de internet del modelo TCP/IP de 4	43
	capas.	
Tabla 4.2.	Protocolos para la capa de aplicación del modelo TCP/IP de 4	44
	capas.	
V. PROTOC	OLO DE TIEMPO REAL PARA LA FUSIÓN DE SENSORES RGE	BD
Tabla 5.1.	Características de la computadora utilizada para la instalación de	67
	OpenNI <sup>®</sup> .	
Tabla 5.2.	Estructura de agrupación de <i>path</i> s.	77
Tabla 5.3.	Ejemplos de movimientos detectados y no detectados por el	79
	módulo de captura.	
Tabla 5.4.	Funciones de los botones del panel frontal.	81
Tabla 5.5.	Características de la tarjeta Raspberry Pi 2, modelo B.	84
VI. RESULT	ADOS Y CONCLUSIONES	
Tabla 6.1.	Cuadro comparativo de los métodos utilizados para vectorización	101
	de imágenes.	
Tabla 6.2.	Comparativa entre los distintos paquetes de software para	102
	vectorización de imágenes.	
Tabla 6.3.	Tiempos promedio para cada etapa del proceso de elaboración	103
	de una marioneta digital.	
Tabla 6.4.	Historial de versiones del módulo de captura implementado en	110
	Processing.	
APÉNDICE	D	
Tabla D1.	Tiempos de diseño para cada una de las etapas del proceso de	255

elaboración de una marioneta digital.

## I. INTRODUCCIÓN

En los últimos años, diversas tecnologías emergentes han dado auge a las interfaces naturales de usuario. Dentro de las principales interfaces naturales de usuario se encuentran las tangibles, de audio y de video. Las interfaces naturales de video permiten operar aplicaciones utilizando sistemas de visión capaces de reconocer gestos y movimientos a partir de la captura de movimiento. Actualmente existen distintos dispositivos y sistemas que facilitan esto, de entre los cuales se pueden mencionar los sensores RGBD que permiten obtener información de color y profundidad. Además, han sido utilizados ampliamente para diversas aplicaciones reduciendo su costo de manera progresiva.

El objetivo del presente trabajo fue utilizar sensores RGBD para la animación en tiempo real de marionetas digitales, es decir, personajes animados que se utilizan en la producción de narrativas digitales para ayudar a estudiantes a comprender un tema o una asignatura, entre otras diversas aplicaciones. El uso de nuevas tecnologías en este tipo de herramientas didácticas puede, como beneficio, fortalecer el aprendizaje de los alumnos, considerando los diferentes estilos de aprendizaje [1]-[3] y promoviendo un aprendizaje colaborativo y creativo [4], [5].

Actualmente existen diversos fabricantes y tipos de sensores RGBD, cada uno con su respectivo SDK (kit de desarrollo de software) y especializado para trabajar sobre una región del cuerpo en específico. En aplicaciones como la animación en tiempo real de marionetas digitales mediante el uso de sensores RGBD, existe un problema no considerado por los actuales SDK que consiste en utilizar distintos sensores RGBD para extraer e integrar las funcionalidades de cada uno de ellos en la misma aplicación. En el presente trabajo se propuso generar una fusión de datos con la información obtenida de la red de sensores RGBD para resolver dicha limitante actual.

Este trabajo se fundamentó en la Plataforma de Fusión de Sensores RGBD para la Animación de Marionetas Digitales (FSA-MD) [6] para la cual se diseñó e implementó un protocolo de red capaz de integrar y fusionar múltiples flujos de datos multimedia capturados desde distintos sensores RGBD, con el propósito de brindar soporte a múltiples clientes multimedia en diversas

1

plataformas y así manipular una marioneta digital por medio de uno o más sensores RGBD de captura de movimiento.

Uno de los rasgos más importantes que se tomó en cuenta para el diseño del protocolo propuesto fue el tipo de información a transmitir. Los sistemas de transmisión de video por fotogramas suelen consumir muchos recursos debido a la alta tasa de paquetes que se transfieren, especialmente en la transmisión de video en tiempo real. En cambio, la ventaja que ofrece el presente protocolo para la plataforma FSA-MD es que no busca enviar cuadros (o *frames*) de video sino un vector de características extraídas de cada sensor. La inclusión del protocolo de fusión de datos en la plataforma FSA-MD permitió producir narrativas digitales, basadas en la animación en tiempo real de marionetas digitales usando múltiples sensores RGBD, logrando hacerlo de una manera práctica, económica y eficiente.

A pesar de que este trabajo de investigación se empleó para un área educativa, puede implementarse en cualquier otra aplicación que requiera el uso de la fusión de datos de una red de sensores RGBD, ya que la principal aportación de este trabajo fue el diseño del protocolo genérico de tiempo real para la fusión de sensores RGBD, independiente de la plataforma de cómputo.

A continuación se describe la estructura general del presente trabajo de tesis:

En el Capítulo II se describen los antecedentes de este trabajo de investigación. De igual manera, se mencionan algunos de los trabajos que se han desarrollado previamente, relacionados con la animación de marionetas digitales mediante sensores RGBD.

En el Capítulo III se muestra el marco teórico, en donde se describen los conceptos generales más relevantes con el fin de contextualizar el presente trabajo de tesis.

En el Capítulo IV se presenta toda la terminología y conceptos principales que fundamentan el diseño de protocolos de red de tiempo real.

2

En el Capítulo V se explica el proceso de diseño del protocolo de red para la fusión de sensores RGBD, así como los elementos que lo conforman y las consideraciones que se tomaron en cuenta para su implementación. También se muestra la validación realizada del protocolo de fusión de sensores RGBD en tiempo real.

En el Capítulo VI se muestran los resultados obtenidos en cada una de las etapas de desarrollo y las conclusiones de este trabajo. Asimismo se presentan algunas recomendaciones y trabajo futuro.

#### **II. ANTECEDENTES**

En este capítulo se presentan diversos trabajos que se han publicado en relación al uso de marionetas digitales para la producción de narrativas digitales, así como la utilización de sensores RGBD en la animación de dichas marionetas. Otros de los trabajos mencionados muestran el uso de protocolos de red de tiempo real para llevar a cabo la fusión de flujos de datos multimedia, así como también se muestran algunas técnicas para realizar esta fusión de datos.

## 2.1 Producción de narrativas digitales

A lo largo de la historia se han empleado diversos métodos y herramientas para producir narrativas digitales. W. Tsou *et al.* [7] desarrollaron el sitio web para narrativas multimedia de la Figura 2.1.



Figura 2.1. Sitio web desarrollado por W. Tsou et al. [7].

Este sitio se compone de 3 módulos: composición de historias multimedia, reproducción de historias y administración de cuentas. A través de estos módulos los usuarios pueden componer o modificar una historia directamente del navegador web. En la historia se pueden agregar fondos y objetos prediseñados, los cuales pueden ser imágenes estáticas o animaciones. También, es posible añadir efectos de sonido, música, grabaciones de voz y texto. Una vez que se ha terminado de componer una historia ésta puede reproducirse en el mismo navegador.

Otra función que ofrece este sitio web es que cada usuario tiene asignada una cuenta, esto permite que cada usuario pueda componer una historia y terminarla o modificarla más tarde simplemente ingresando a su cuenta.

Años más tarde O. Balet introdujo INSCAPE [8], una plataforma que cuenta con una colección de herramientas que permite la creación intuitiva de historias interactivas. INSCAPE cuenta con módulos de adquisición que pueden utilizarse para reconstruir ambientes virtuales, ya sea a partir de imágenes panorámicas o de la captura de movimientos provenientes de películas con artistas reales. INSCAPE también tiene módulos de edición que permiten la creación y configuración de elementos 2D o 3D para la historia, tal como personajes, accesorios, fondos y sus respectivas animaciones. Con INSCAPE es posible crear y manipular diálogos hablados, música y efectos sonoros. En la Figura 2.2 se muestra la interfaz gráfica de INSCAPE.



Figura 2.2. INSCAPE versión beta [8].

A raíz de INSCAPE surgieron otros trabajos como el de S. Göbel *et al.* [9] quienes presentaron una plataforma para narrativas digitales interactivas denominada StoryTec. Esta plataforma tiene dos componentes principales, una plataforma de creación de historias y un motor de tiempos de ejecución. Por medio del editor de historias de StoryTec se obtiene una representación interactiva 2D de la historia.

Otro de los componentes de StoryTec es el editor de escenas con el cual es posible crear escenarios 2D o 3D insertando objetos desde una librería, tal como se muestra en la Figura 2.3.



Figura 2.3. Editor de escenarios para juegos de aventura [9].

Una desventaja de estos trabajos es que no cuentan con alguna opción en la que los usuarios puedan utilizar sus propios fondos y objetos sino que deben seleccionarlos de una librería ya existente. A deferencia de estos, sistemas como iTell [10], ShadowStory [11], Props [12], MOVIO [13] y PowToon [14] permiten que los usuarios creen o importen sus propios recursos digitales, tales como personajes, objetos, fondos y sonidos. En particular, PowToon ha sido utilizado ampliamente para la creación de presentaciones digitales, ya que se ha apoyado del concepto de narrativas digitales para crear presentaciones en video.

La producción de narrativas digitales no está limitada a plataformas PC o web ya que también se han diseñado e implementado herramientas para dispositivos móviles. Por ejemplo, J. Multisilta y M. Suominen desarrollaron una herramienta para dispositivos móviles llamada MoViE (*Mobile Video Experience*) [15], la cual se muestra en la Figura 2.4. MoViE está diseñada para producir historias en video utilizando la cámara de los dispositivos móviles, una vez que los usuarios graban el video y lo suben a la plataforma MoViE tienen la posibilidad de etiquetarlo y hacer remezclas con él. Como parte de las funciones de MoViE también se pueden añadir etiquetas geográficas utilizando los datos del GPS (*Global Positioning System*) del

dispositivo. Aunque MoViE ofrece ciertos beneficios, tiene como desventaja que no permite compartir los videos en la red en tiempo real.

MOVIE		+-
[Frontpage] [ [Groups] [Upl [Moderate]	<u>Logout]</u> jari oad] <u>[Remix]</u> [	<u>Settings]</u>
Change order	to: <u>Popular</u>	0
* Search word	:	Send
Order	Match	•
Latest videos 1/36 [ <u>&gt;]</u> [ <u>&gt; </u> ]		
[ <u>Source</u> ] Sum witcher 2009,	<b>uinen koost</b> /05/06 13:27	e (0:40)
	Kolme su yyteriä ed	muista litoitu
Valinnat		Takaisi

Figura 2.4. Página de inicio de MoViE en un celular Nokia N95 [15].

Otro trabajo es el de D. Frohlich *et al.* quienes desarrollaron StoryBank [16], un sistema que permite crear historias audiovisuales utilizando la cámara de un dispositivo móvil. Por cada historia se admiten hasta 6 imágenes capturadas con el dispositivo móvil y una pista de audio con una duración de hasta 2 minutos. Los usuarios pueden grabar los elementos de la historia (audio o video) y después trabajar en la sincronización. Una vez que se termina el proceso de producción de una historia se puede subir al repositorio de StoryBank. De igual forma, las historias del repositorio pueden descargarse a un dispositivo móvil. Para ambos casos, la transferencia de las historias se hace vía *Bluetooth*. Aunque las historias almacenadas en el servidor pueden ser visualizadas por otros usuarios, no se pueden visualizar en tiempo real, ya que cada historia debe estar previamente almacenada en el repositorio para poder reproducirse después.

Además de estos trabajos, existen algunas aplicaciones para dispositivos móviles como Storie<sup>®</sup> [17], Story Dice<sup>®</sup> [18], [19], Steller<sup>®</sup> [20] y Tellagami<sup>®</sup> [21], entre otras [22]-[30].

7

Una de estas aplicaciones es Episodium (Figura 2.5), desarrollada en el Laboratorio de Aprendizaje Móvil del I. T. de Chihuahua por A. Pacheco [31].



Figura 2.5. Aplicación móvil Episodium [31].

Esta aplicación utiliza la captura de fotos y grabación de voz para producir episodios, una vez que se tienen los episodios terminados se pueden reproducir como presentaciones audiovisuales e incluso se pueden compartir, también permite que los usuarios puedan modificar o borrar cualquier episodio. Episodium tiene como desventaja que solo está disponible para la plataforma iOS, además solo soporta 32 fotos por cada episodio y 64 episodios en total. Otro aspecto importante es que el enfoque de esta aplicación está en el uso de fotografías, por lo que no es posible insertar algún tipo de elemento externo, como un fondo o archivo de audio.

#### 2.2 Elaboración de marionetas digitales

Un recurso que puede ser útil al momento de producir una narrativa digital son las marionetas digitales y para elaborar una marioneta digital se pueden utilizar diversas herramientas. En la actualidad existen algunos paquetes de software para el diseño de gráficos por computadora con los cuales se pueden crear personajes animados para utilizarlos como marionetas digitales. Algunos de los programas más utilizados son Adobe Photoshop<sup>®</sup> [32], Adobe Illustrator<sup>®</sup> [33],

CorelDraw<sup>®</sup> [34] y Autodesk<sup>®</sup> Graphic [35]. En la Figura 2.6 se muestra un dibujo hecho con Adobe Illustrator<sup>®</sup>.



Figura 2.6. Personaje trazado en Adobe Illustrator®.

El inconveniente que tienen muchos de estos programas es que su uso requiere de una licencia, que regularmente es muy costosa, dejando a muchos usuarios sin la posibilidad de utilizar estos paquetes computacionales.

Como alternativa a lo anterior, existen algunos programas de libre uso, tal es el caso de Gimp<sup>®</sup> [36], Inkscape<sup>®</sup> [37], Skencil<sup>®</sup> [38] y Xara LX<sup>®</sup> [39], entre otros. No obstante, otra desventaja de algunas de estas herramientas de software, ya sean de libre uso o no, es que no se pueden utilizar en distintas plataformas sino que en algunos casos son diseñadas para trabajar solo en una de ellas, como Autodesk<sup>®</sup> Graphic que es exclusiva de los sistemas operativos de Apple<sup>®</sup> (iOS y Mac OS) o Xara LX<sup>®</sup> que existe solo para Linux, esto limita fuertemente su uso en usuarios que no cuentan con determinado sistema operativo o dispositivo. Otro aspecto a considerar al momento de elegir una herramienta de trabajo es la facilidad de uso. Muchos de los programas de diseño existentes requieren que el usuario tenga una amplia experiencia en su uso, sin embargo, para el diseño y elaboración de una marioneta digital no siempre se cuenta con un diseñador o usuario experto. Por este motivo, algunas personas han optado por dibujar

de manera manual (a lápiz y papel) sus propios personajes y digitalizarlos utilizando un escáner (Figura 2.7).



Figura 2.7. Dibujo hecho a lápiz y papel: a) foto real del dibujo, b) dibujo digitalizado.

Esta última forma de trabajo trae consigo otros desafíos a resolver, como la posibilidad de escalar el dibujo o realizar modificaciones en él, la calidad de bordes y colores, entre otros.

Los paquetes de software mencionados anteriormente facilitan la creación de marionetas digitales pero solamente lo hacen para un plano 2D, aunque los personajes puedan tener una perspectiva de diseño tridimensional (Figura 2.8), su visualización solo puede ser en un plano 2D.



Figura 2.8. Dibujo 2D visto desde diferentes perspectivas [40].

En algunos casos lo que se requiere es una marioneta digital 3D, para esto es necesario utilizar otro tipo de herramientas (véase 2.3.4).

En cuanto a los gráficos 2D, entre ellos las marionetas digitales, existen diversos tipos de formatos con los cuales se puede trabajar. Principalmente, los gráficos pueden ser mapas de bits (véase 3.4.1) o gráficos vectoriales (véase 3.4.2).

La mayoría de los programas mencionados anteriormente permiten la creación de gráficos vectoriales, facilitando el proceso de animación de dichos gráficos, es por ello que algunos trabajos [41]-[44] utilizan marionetas digitales 2D basadas en este tipo de gráficos.

#### 2.3 Técnicas para la animación de marionetas digitales

A lo largo de los años se han desarrollado un sinnúmero de técnicas para la creación de animaciones. En la Figura 2.9 se muestra una propuesta para la clasificación de estas técnicas, tomando como base los trabajos de J. Gutiérrez [45] y P. Domínguez [46]. Cabe mencionar que no todas estas técnicas se emplean en la animación de marionetas digitales. A continuación se presentan las técnicas más utilizadas.



Figura 2.9. Clasificación general de las técnicas de animación.

#### 2.3.1 Animación mediante la técnica stop motion

La técnica de *stop motion* consiste en tomar una fotografía de un personaje u objeto tridimensional por cada *frame* de animación [45], [47]. Como se aprecia en la Figura 2.10, *stop motion* tiene distintas variantes, de las cuales una de ellas es la animación mediante recortes.



Figura 2.10. Clasificación de las técnicas de animación tradicional.

C. Barnes *et al.* presentaron una interfaz, basada en video, que permite crear animaciones mediante recortes a través de los movimientos de los personajes [48], para esto, los usuarios crean los recortes físicamente y por medio de la interfaz se crea una animación en tiempo real conforme se van moviendo cada uno de los recortes. La diferencia de la técnica utilizada en esta interfaz con el *stop motion* tradicional es que no se toma una fotografía por cada *frame* de animación, sino que cada *frame* se puede visualizar en tiempo real gracias al uso de la captura de movimiento. Así como éste, se pueden encontrar otros tipos de trabajos relacionados a la técnica de *stop motion* como [49]-[52].

## 2.3.2 Animación 2D

Como se muestra en la Figura 2.11, la animación 2D se puede realizar por medio de diferentes técnicas.



Figura 2.11. Clasificación de las técnicas de animación digital.

En trabajos como el de S. Piman y A. Talib [53] se utiliza la animación flash para manipular una marioneta digital. Para esto, la marioneta debe estar seccionada, tomando en cuenta cuáles puntos se utilizarán para articular la marioneta. Otro tipo de técnica es la animación por fotogramas clave (en inglés, *keyframe*), en esta técnica se definen posiciones clave en puntos específicos en el tiempo [54], para completar la animación se puede utilizar la interpolación de fotogramas clave o la animación basada en *spline* [55].

Actualmente existen numerosas herramientas para la animación 2D, algunas de estas herramientas son Moho<sup>®</sup> [56], Harmony<sup>®</sup> [57], Synfig Studio<sup>®</sup> [58], Adobe After Effects<sup>®</sup> [59], Adobe Animate<sup>®</sup> [60] y Clip Studio Paint<sup>®</sup> [61], entre otras.

## 2.3.3 Animación mediante la captura de movimiento

La animación de marionetas digitales puede realizarse mediante fotogramas clave, sin embargo dicho método es tardado y complejo [41], [42], es por ello que se ha introducido la

captura de movimiento para la animación de marionetas digitales [5]. Por medio de esta técnica es posible reducir el tiempo utilizado en la producción de la animación, ya que la animación se lleva a cabo en tiempo real, es decir, de acuerdo a los movimientos detectados y registrados se produce una animación en la marioneta digital en tiempo real.

Existen diversos trabajos que utilizan la captura de movimiento para la animación de marionetas digitales, algunos de ellos emplean la realidad dual manipulando los movimientos de una marioneta digital por medio de marionetas físicas. Por ejemplo, M. Antonijoan *et al.* [43] desarrollaron un sistema híbrido en el que es posible manipular avatares en un mundo virtual por medio de marionetas físicas, las cuales actúan como elementos de una interfaz tangible. Por otro lado, A. Mazalek y M. Nitsche [62] presentaron un sistema en donde utilizan sensores para captar el movimiento de una marioneta física y reproducirlo en tiempo real en una marioneta digital. En el trabajo de N. Hiroki *et al.* [63] se utiliza una marioneta con 10 marcas ARToolKit colocadas en ella, cuando la marioneta está en movimiento se captura la posición de las marcas con una cámara para finalmente determinar cuáles articulaciones fueron afectadas de forma visible durante la animación utilizando una base de datos de movimientos.

Con la aparición de los sensores RGBD los trabajos relacionados con la captura de movimiento han ido en aumento. Algunos de esos trabajos se han enfocado en áreas como medicina [64], [65], robótica [66], [67], control de vehículos no tripulados [68] y educación, entre otras.

Tal y como se propone en algunos trabajos una persona puede fungir como el controlador de una marioneta digital. Por medio de la captura de movimiento la persona que protagoniza la animación puede ver en tiempo real los movimientos que va realizando, además la interacción con la marioneta digital se vuelve más fluida y simple [42].

De los diversos trabajos que existen para el control de marionetas digitales, algunos se enfocan en la manipulación de marionetas digitales 3D, creando avatares en ambientes virtuales [44], [69], [70]. Muchos de estos trabajos se apoyan de herramientas de software como Unity<sup>®</sup>

[71], la cual permite trabajar con representaciones 3D y es ampliamente utilizada para la creación de animaciones en videojuegos.

De la misma manera existen trabajos centrados en animaciones 2D [72], [73]. En el trabajo presentado por L. Leite y V. Orvalho [42] se propuso un sistema para estudiar la interacción entre artistas no expertos y marionetas virtuales en 2D y 3D, los resultados de dicho trabajo mostraron que con pocos movimientos los usuarios pudieron manipular las animaciones 2D, sin embargo prefirieron interactuar con las marionetas 3D.

Otro de los trabajos relacionados se desarrolló en el Laboratorio de Aprendizaje Móvil del I. T. de Chihuahua [5], [73], éste cuenta con un sistema de captura de movimiento compuesto por una cámara de profundidad que adquiere información RGBD, y un reproductor de animación de marionetas implementado en Processing (Figura 2.12).



Figura 2.12. Arquitectura inicial del sistema de animación de marionetas en tiempo real para un solo sensor RGBD [6], [73].

## 2.3.4 Animación 3D

Como se mencionó en la sección anterior, las marionetas digitales también pueden estar hechas a base de gráficos 3D, para estos casos es necesario utilizar herramientas de software que permitan crear modelos tridimensionales, dentro de éstas se encuentran Unity<sup>®</sup>, Blender<sup>®</sup> [74], Maya<sup>®</sup> [75] y Motion Builder<sup>®</sup> [76], entre otras. Las técnicas que se utilizan para animar marionetas digitales 3D se apoyan mayormente de la captura de movimiento, para producir las animaciones algunos trabajos capturan el movimiento del cuerpo de una persona mediante sensores RGBD [77], otros usan sistemas ópticos como OptiTrack<sup>®</sup> para capturar el movimiento de marcas en objetos (como guantes) [78], [79] y otros se aprovechan de las interfaces naturales de usuario para crear la animación 3D [80].

En la actualidad han surgido sistemas de realidad virtual, por lo que se ha ido integrando esta tecnología en la animación de personajes digitales, un ejemplo de esto es el trabajo presentado por C. Curtis *et al.* [81] quienes hicieron la producción videográfica Pearl utilizando realidad virtual.

#### 2.4 Transmisión de flujos de datos multimedia en tiempo real

Con el avance y creación de nuevas tecnologías web, la transmisión de información multimedia ha tomado mayor relevancia [82], [83], en gran parte esto se debe al creciente uso de dispositivos móviles. En los últimos años se han desarrollado algunas aplicaciones que han revolucionado la comunicación entre personas, tal es el caso de Skype<sup>®</sup> (2003) [84], WhatsApp<sup>®</sup> (2010) [85], Facebook<sup>®</sup> Chat (2008), Facebook<sup>®</sup> Messenger (2011) [86] y FaceTime<sup>®</sup> (2011) [87], entre otras. Recientemente han salido al mercado otras nuevas aplicaciones como las de Google<sup>®</sup> Duo y Allo [88], [89], por medio de estas tecnologías es posible realizar el intercambio de mensajes de texto, audio, video o archivos, todo esto en tiempo real.

Debido a la reciente aparición de aplicaciones para dispositivos móviles como YouNow<sup>®</sup> (2011) [90], Periscope<sup>®</sup> (2015) [91] y Facebook<sup>®</sup> Live (2016) [86] ha sido posible transmitir video en vivo, esto es, en tiempo real, gracias a ello se han aprovechado dichas tecnologías para desarrollar nuevas aplicaciones y sistemas enfocados a diversas áreas. Por ejemplo, A. Alohali *et al.* diseñaron una herramienta para corredores, que permite producir narrativas de modo que el usuario pueda compartir su experiencia mientras se ejercita, esto con el fin de motivar al usuario a correr más o a ser más activo [92]. En este sistema se utilizó una cámara GoPro<sup>®</sup> y la aplicación Periscope<sup>®</sup> para la transmisión de video.
La transmisión de flujos de datos multimedia también implica el envío de audio. En el trabajo de J. Xu [93] se diseñó un sistema para generar de manera automática una animación de baile sincronizada con una pieza musical. Para los movimientos de la animación se utilizó la base de datos de captura de movimiento de la Universidad Carnegie Mellon (CMU) y a partir de ésta se extrajeron gráficos de movimiento. En cuanto a la pieza musical, se utilizaron dos escenarios para su sincronización, transmisión (*streaming*) y no transmisión (*non-streaming*), para el primero se utilizó un esquema de doble almacenamiento *buffer*.

Dentro del área de transmisión de datos multimedia también se puede considerar la transmisión de la información adquirida mediante la captura de movimiento. En el trabajo desarrollado por M. Patoli *et al.* [70] se presentó eMove, un sistema en línea de captura de movimiento de tiempo real enfocado en aplicaciones de entretenimiento. Una de las aplicaciones de eMove es eMoveChat, la cual es una aplicación de red cliente-servidor con una comunicación uno a uno, que tiene la finalidad de transmitir datos de audio y video de una animación generada mediante la captura de movimiento en tiempo real de una persona. eMoveChat trabaja con una aplicación en línea de avatares 3D, donde dichos avatares interactúan con la gente en el mundo real. El modo de comunicación para el envío de audio y texto es bidireccional mientras que para video es unidireccional. Uno de los mayores retos de este trabajo fue sincronizar la animación facial de los avatares con la voz.

#### 2.5 Redes de tiempo real de sensores RGBD

Existen algunos trabajos que buscan integrar y fusionar los datos adquiridos por varios sensores RGBD para generar un solo flujo de datos de salida. Uno de estos trabajos pertenece a M. Uchinoumi *et al.* [94] quienes propusieron un sistema de captura de movimiento en tiempo real para obtener la reconstrucción tridimensional de una persona en movimiento, para ello utilizó varias cámaras en un sistema cliente-servidor, cada una de ellas asociada a un cliente. Por cada *frame*, cada uno de los clientes produce una imagen de la silueta de la persona, la cual se proyecta en el fondo de un espacio tridimensional. El resultado de esta proyección se envía al servidor en cada tiempo de muestreo y éste genera un área común tomando como base todos los fondos proyectados.

F. Cassola *et al.* presentaron un gimnasio virtual en línea utilizando varios sensores RGBD [95], [96], en la Figura 2.13 se muestra este sistema.



Figura 2.13. Escenario del gimnasio en línea propuesto por F. Cassola et al. [95].

En este gimnasio en línea se utilizaron múltiples sensores Kinect<sup>®</sup> conectados en red para enviar a un servidor central los datos adquiridos por cada uno de ellos. La idea central de este trabajo fue conectar a diversos usuarios ubicados en distintas locaciones, a una misma sesión de entrenamiento dirigida por un instructor. Para representar de manera visual a cada uno de los usuarios y al instructor se diseñaron avatares 3D con sus respectivos escenarios virtuales, por medio de cada sensor Kinect<sup>®</sup> se capturó el movimiento del usuario reproduciéndolo en tiempo real en el avatar. Otra característica importante de este sistema es que permite la interacción entre todos los participantes de la sesión. Como resultado de la implementación de este sistema se obtiene una sesión de entrenamiento en donde el instructor puede visualizar en tiempo real los movimientos de un usuario, y al mismo tiempo los usuarios pueden ver los movimientos del instructor.

## **III. MARCO TEÓRICO**

A continuación se presentan los conceptos básicos que respaldan este trabajo de tesis con el objetivo de crear un contexto adecuado para brindar una mejor comprensión sobre éste. En este capítulo se abordan temas de las siguientes áreas: narratología, animación digital, interfaces de usuario, percepción y aplicaciones multimedia.

## 3.1 Narratología

La narratología es la ciencia de la narrativa, la cual estudia su forma y funcionalidad así como las similitudes y diferencias existentes entre ellas [97], [98], también ayuda a comprender, analizar y evaluar una narrativa [99].

A partir de la narratología se derivan otros términos relacionados, lo cuales se describen a continuación.

## 3.1.1 Narrativa

De acuerdo a la narratología, una narrativa es la representación semiótica<sup>1</sup> de una serie de eventos conectados significativamente de manera causal y temporal [98], [100], algunos ejemplos de narrativas son: películas cinematográficas, obras teatrales, historietas, novelas, noticiarios, diarios, crónicas y tratados de historia geológica [98]. Las narrativas pueden construirse utilizando diferentes medios semióticos como lenguaje escrito o hablado, imágenes visuales, gestos, actuaciones o la combinación de ellos [98], [101]. Una narrativa se puede clasificar como ficticia, no ficticia, verbal, visual o verbal-visual [102].

Dado que una narrativa es una representación semiótica de una serie de eventos, un nivel de análisis examinará los eventos representados mientras que otro nivel examinará la estructura de la representación. Los niveles de análisis varían entre los distintos autores, algunos distinguen 2 mientras que otros hablan de 3 o 4 [98]. Según M. Bal, el análisis de narrativas parte de tres niveles: fábula, historia y texto [99], tal y como se muestra en la Figura 3.1.

<sup>&</sup>lt;sup>1</sup> Ciencia que lleva a cabo el estudio de los signos en la vida social [199].



Figura 3.1. Representación de los niveles de análisis de una narrativa.

### 3.1.2 Narrativa textual

Para comprender qué es una narrativa textual se debe definir primeramente qué es un texto. Un **texto** es una construcción lingüística [98], una estructura finita compuesta por un lenguaje de signos [99]. Con base en esto se tiene que una narrativa textual es un texto en el cual un agente relata una historia en un medio en particular como en un escrito, imágenes o sonidos [99]. Asimismo una narrativa textual puede ser lingüística, teatral, pictórica o fílmica [98].

### 3.1.3 Historia

Es el significado de un texto narrativo, un esquema cognitivo de eventos [98] en donde se entiende un evento como la transición de un estado a otro [99]. Una historia es también una secuencia de eventos que tiene un principio, una parte intermedia y un final. Regularmente las historias generan suspenso debido a las complicaciones presentadas en la parte intermedia, las cuales son resueltas al terminar la historia [102].

# 3.1.4 Fábula

Es una serie de eventos en la cual intervienen actores, dichos eventos están lógica y cronológicamente relacionados. Por su parte, los actores son quienes ejecutan las acciones y no necesariamente tienen que ser personas [99]. A diferencia de la historia, la fábula se centra más en la acción mientras que la historia se enfoca en cómo se transmite dicha acción [98].

#### 3.2 Narrativas digitales

En términos prácticos las narrativas se pueden definir como un proceso natural en el que se comparten historias, ideas o experiencias a través de palabras y acciones [103]-[105]. Las narrativas son herramientas prácticas y de gran alcance que se pueden utilizar para la enseñanza, debido a que por medio de éstas, los estudiantes pueden personalizar los conocimientos adquiridos y construir su propio aprendizaje de acuerdo a las historias que hayan escuchado o contado [103], [105].

Con el fin de lograr que las narraciones sean más interactivas, se han desarrollado las narrativas digitales, las cuales han surgido como una herramienta educativa que maneja el arte de relatar historias utilizando elementos multimedia tales como imágenes, audios y videos [103], [106], [107]. Mediante el uso de este recurso los estudiantes se envuelven en un aprendizaje colaborativo, significativo y profundo [4], debido a que el diseño e implementación de una narrativa digital permite a los estudiantes desarrollar distintas habilidades como la investigación, escritura, organización, manejo de diversas tecnologías, expresión oral y solución de problemas, entre otras [108], [109].

#### 3.3 Marionetas digitales

El uso de marionetas puede ser una herramienta educativa muy efectiva, en especial si se utiliza en narraciones [41]. Las marionetas digitales, como agentes pedagógicos, son personajes animados que ayudan a los estudiantes a comprender una asignatura [110]. Dentro de las narrativas digitales, las marionetas digitales se han utilizado como una herramienta para la producción de historias [42], esto favorece el proceso de la narración ya que por medio de personajes digitales animados se pueden establecer diálogos simulando la interacción entre dos o más personas.

Uno de los rasgos que se debe tomar en cuenta para el diseño de marionetas digitales es el tipo de gráficos con el que éstas son elaboradas, para ello existen diferentes tipos de gráficos.

#### 3.4 Gráficos por computadora

A lo largo de la literatura se pueden encontrar muchas definiciones para los gráficos por computadora. De forma general, los gráficos por computadora se pueden definir como el proceso de producir una foto o imagen utilizando una computadora [111]. Para representar la información de un gráfico en una computadora existen dos sistemas gráficos principales, los gráficos rasterizados (mapas de bits) y los gráficos vectoriales [112].

#### 3.4.1 Gráficos de mapas de bits

En este tipo de gráficos, una imagen está compuesta por una colección de pixeles (mapa de bits) [112]-[114]. Un pixel es el elemento más simple en los gráficos por computadora y representa una posición en una imagen (Figura 3.2), además, se puede tener control sobre el pixel estableciendo su intensidad y color [113].



Figura 3.2. Arreglo de pixeles.

Existen algunos procedimientos para determinar cuál pixel da la mejor aproximación a la imagen u objeto gráfico deseado. Este proceso de determinación de los pixeles apropiados para representar una imagen (u objetos gráficos) se conoce como rasterización. Por otro lado, el proceso de representar continuamente una imagen (u objetos gráficos) como una colección de pixeles discretos se conoce como conversión de barrido [113].

Cuando una imagen no tiene color se le llama imagen en escala de grises (Figura 3.3), en éstas cada pixel tiene un valor que indica el nivel de gris de dicho pixel.



Figura 3.3. Ejemplo de una imagen en escala de grises: a) imagen original, b) imagen en escala de grises, c) niveles de gris de una porción de la imagen.

En una imagen de color, el valor de cada pixel usa 3 valores para almacenar el rango de intensidades de los componentes verde, rojo y azul (RGB) [111], [114], ver Figura 3.4.



Figura 3.4. Componentes RGB de una imagen: a) imagen original, b) componente rojo, c) componente verde, d) componente azul.

Las principales desventajas que presentan los gráficos de mapas de bits son la pérdida de resolución al momento de escalarse y su contenido estático (no interactivo). Si una imagen se escala a un tamaño más grande que la original, ésta sufrirá una degradación en su calidad ya que estos gráficos tienen una resolución fija, ver Figura 3.5. En cuanto a la interactividad, si bien existen algunos formatos para animación de imágenes, no permiten modificar de una manera dinámica su contenido [114].



Figura 3.5. Gráfico de mapa de bits escalado: a) imagen original, b) imagen escalada.

Anteriormente todos los gráficos que se visualizaban en la web eran imágenes de mapas de bits en formatos como GIF (*Graphics Interchange Format*) o JPEG (*Joint Photographic Experts Group*), no obstante, en la actualidad también se utilizan otros tipos de gráficos como los vectoriales [114].

## 3.4.2 Gráficos vectoriales

Las imágenes vectoriales están basadas en fórmulas matemáticas, vectores y sistemas de coordenadas, gracias a esto no existen limitantes sobre el tamaño del pixel. Debido a lo anterior, es posible modificar la resolución del gráfico para cualquier tamaño de la imagen [115] (Figura 3.6) sin que ésta pierda calidad.

Algunos aspectos importantes que se deben tomar en cuenta al trabajar con gráficos vectoriales son el sistema de coordenadas en el cual se genera el gráfico (véase 3.4.3) y las transformaciones 2D (véase 3.4.4) que se puedan aplicar al mismo.



Figura 3.6. Comparación entre gráficos: a) gráfico de mapa de bits, b) gráfico de mapa de bits escalado, c) gráfico vectorizado.

# 3.4.3 Sistema de coordenadas de pixeles

En un sistema de coordenadas cartesianas bidimensionales el origen 0 se encuentra en las coordenadas (0,0), punto en el que se intersectan los ejes x y y (Figura 3.7).



Figura 3.7. Sistema de coordenadas cartesianas bidimensionales.

La resolución de una pantalla o el tamaño de una imagen, por lo regular, se especifica en pixeles, cada uno de los cuales está ubicado en cierta posición (x, y). A diferencia de las coordenadas cartesianas bidimensionales, en las coordenadas de pixeles se toma como origen la esquina superior izquierda de la imagen o pantalla (Figura 3.8) [116].



Figura 3.8. Sistema de coordenadas de pixeles.

## 3.4.4 Transformaciones 2D

Dado que un gráfico vectorial está conformado por vectores se pueden aprovechar las transformaciones lineales para modificar el gráfico 2D. Existen diferentes tipos de transformaciones que se pueden aplicar a un gráfico, siendo las más relevantes la traslación, rotación, escala, reflexión e inclinación (Figura 3.9).



Figura 3.9. Diferentes tipos de transformaciones 2D: a) gráfico original, b) reflexión, c) rotación, d) inclinación, e) escala, f) traslación.

## 3.5 SVG

SVG (*Scalable Vector Graphics*, Gráficos Vectoriales Redimensionables) es un formato para gráficos vectoriales que utiliza XML (*eXtensible Markup Language*, Lenguaje de Marcado Extensible), esto permite que los gráficos puedan visualizarse fácilmente en un navegador web [112]. En el Anexo 1 se muestra una tabla de compatibilidad de los navegadores actuales con SVG.

## 3.5.1 Estructura de un documento SVG

Como se muestra en la Figura 3.10, para definir un gráfico SVG en un documento XML se debe seguir una determinada estructura. En la primer línea se debe especificar la versión XML que se está utilizando, en segundo lugar se debe incluir el tipo de documento mediante la etiqueta **<!DOCTYPE>**. Con estas dos líneas el documento ya puede ser reconocido por los navegadores como un documento SVG. A continuación, toda la especificación SVG se debe situar dentro de las etiquetas **<svg> y </svg>**.

Una de las propiedades que deben establecerse siempre en un documento SVG son las dimensiones del gráfico, éstas se definen por medio de los atributos width y height.

```
<?xml version="1.0" encoding="UTF-8"?>
 1
    <!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
 2
 3
       "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
 4
    <svg
 5
      width="400"
      height="400"
 6
      xmlns="http://www.w3.org/2000/svg">
 7
      <title>Gráfico</title>
 8
      <desc>Este es un gráfico de ejemplo</desc>
 9
      <!-- La definición del gráfico va aquí -->
10
    </svg>
11
```



# 3.5.2 Formas básicas

Los gráficos vectoriales se generan a partir de formas básicas. En la Tabla 3.1 se muestran las diferentes formas básicas que se pueden utilizar para crear gráficos vectoriales en SVG. Cada uno de los elementos de SVG tiene ciertas propiedades que deben definirse para que éstos se muestren apropiadamente en el navegador web, dichas propiedades son la posición, dimensión, color y contorno.

Forma	Etiqueta SVG	Ejemplo	
Rectángulo	<rect></rect>	<rect <br="" height="100" width="300" x="100" y="100">style="stroke: black; fill:none" /&gt;</rect>	
Círculo	<circle></circle>	<circle <="" cx="200" cy="200" r="150" th=""></circle>	
		<pre>style="stroke: black; fill: none"/&gt;</pre>	
Elipse	<ellipse></ellipse>	<ellipse <="" cx="200" cy="200" rx="150" ry="50" th=""></ellipse>	
		<pre>style="stroke: black; fill:none" /&gt;</pre>	

Línea	<line></line>	<li>x1="50" y1="50" x2="350" y2="350"</li>	
		<pre>style="stroke: black" /&gt;</pre>	
Poli-línea	<polyline></polyline>	<polyline <br="" points="50,50 150,100 250,100 350,150">style="stroke: black; fill: none" /&gt;</polyline>	
Polígono	<polygon></polygon>	<pre><polygon <="" points="50,50 150,150 250,150 350,250" pre=""></polygon></pre>	
		Style= Stroke. Dlack, Till. Hone //	
Trayecto	<path></path>	<pre><path black;="" d="M 286.3003,113.80364 A 106.06602,99.500023&lt;/th&gt;&lt;/tr&gt;&lt;tr&gt;&lt;th&gt;(path)&lt;/th&gt;&lt;th rowspan=2&gt;&lt;/th&gt;&lt;th&gt;&lt;pre&gt;style=" fill:="" none"="" stroke:=""></path></pre>	

#### 3.6 Interacción hombre-máquina

La interacción hombre-máquina (HCI) se encuentra en la intersección de las ciencias sociales, ciencias del comportamiento y la tecnología de computación e información [117]. HCI se ocupa de la teoría, diseño, implementación y evaluación de las formas en que las personas utilizan e interactúan con los equipos de cómputo [118], [119].

Existen cuatro aspectos principales en los que se enfoca la interacción hombre-máquina, los cuales son: personas, computadoras, tareas a realizar y usabilidad. En HCI las personas o usuarios son quienes deben ser asistidos por los sistemas computacionales, para ello es importante considerar los siguientes aspectos: cómo perciben el mundo a su alrededor, cómo almacenan y procesan la información así como solución de problemas, y finalmente, cómo manipulan físicamente los objetos. La interacción de una persona con su alrededor se lleva a cabo a través del envío (salida) y recepción (entrada) de la información. Como se muestra en la Figura 3.11, en HCI un usuario recibe información de la computadora y éste responde enviando información de vuelta a la computadora [119].



Figura 3.11. Flujo de información en HCI.

## 3.7 Interacción multimodal

La interacción multimodal se refiere a la capacidad para utilizar diferentes formas o modos de comunicación al momento de suministrar y obtener información de un servicio [120], [121]. Para que la interacción multimodal sea natural, los canales de comunicación entre la persona y el ambiente deben ser los mismos que los canales que utilizan las personas para comunicarse entre sí [120]. La multimodalidad es un proceso en donde los dispositivos y el usuario son capaces de formar una interacción conjunta desde cualquier sitio y en cualquier momento [121].

En una interacción multimodal, los sistemas pueden procesar métodos de entrada naturales combinados como voz, lápiz, táctil, gestos de la mano y movimientos corporales, entre otros [122], [123], el usuario puede determinar el modo o modos de interacción que desea utilizar para acceder a la información [121]. De la misma manera, un sistema multimodal puede emplear varios canales de salida (por ejemplo, voz, texto, gráficos o imágenes) para proporcionar información al usuario y estimular varios sentidos del usuario de forma simultánea [122].

Según el consorcio *World Wide Web* (W3C) existen tres tipos de canales de comunicación implicados en la entrada de datos en la multimodalidad: el secuencial, simultáneo y compuesto [121].

#### 3.8 Interfaces de usuario

De acuerdo a S. Daniel [124] y J. Yim [125], las interfaces de usuario (UI) se pueden clasificar como se muestra en la Figura 3.12.



Figura 3.12. Clasificación de las interfaces de usuario.

Las interfaces de línea de comandos (CLI) se basan en texto y requieren que el usuario recuerde los comandos exactos que debe introducir en el teclado [124]-[126], por otro lado, las interfaces gráficas de usuario (GUI) se basan en gráficos y permiten que los usuarios interactúen con la computadora mediante íconos [126], [125]. Las interfaces naturales de usuario (NUI) se basan en gestos y movimientos, mientras que las interfaces orgánicas de usuario (OUI) [127]-[129] se basan en los sistemas orgánicos.

#### 3.9 Interfaces naturales de usuario

Una NUI es un tipo de HCI basada en el análisis automático del comportamiento natural del usuario, lo que permite que los usuarios interactúen con la interfaz en un ambiente natural [130], [131]. El comportamiento natural de una persona se refiere a las actividades que ésta realiza en la vida diaria para interactuar con su ambiente. Los usos más comunes del comportamiento natural en los sistemas computacionales son el seguimiento de ojos, identificación de expresiones faciales, interacción basada en gestos mediante interfaces multitáctiles, análisis de emociones e interfaces de voz, entre otros [131].

En la Figura 3.13 se presentan tres tipos básicos de interfaces naturales de usuario, las interfaces tangibles, interfaces de visión y las interfaces de audio. Esta clasificación está basada de acuerdo a los diferentes sentidos del ser humano y su correspondencia a cada grupo en particular. Los sentidos del gusto y olfato no se pueden ubicar dentro de las interfaces naturales de usuario ya que actualmente no es posible utilizarlos en la interacción hombre-máquina [131].



Figura 3.13. Clasificación de las NUI de acuerdo a los 5 sentidos del ser humano.

## 3.9.1 Interfaces de usuario tangibles

Las interfaces de usuario tangibles (TUI) han surgido como un nuevo estilo de interacción que enlaza el mundo físico con el digital. Las TUI se caracterizan por utilizar objetos físicos para representar y manipular datos [132], lo que promueve la operación intuitiva de aplicaciones informáticas [133]. Es importante resaltar que un sistema que utiliza dispositivos físicos no necesariamente es una TUI, ya que el término físico no es un sinónimo de tangible, un objeto físico llega a ser tangible solo cuando éste representa o manipula información digital [134].

## 3.9.2 Interfaces de usuario de audio

El audio puede ser utilizado en una interfaz de usuario para el control de aplicaciones, sin embargo, requiere de un análisis de voz [135]. Las interfaces de usuario de audio (AUI) o interfaces de voz permiten la interacción por medio de la comunicación mediante una aplicación de lenguaje hablado. Los elementos de este tipo de interfaces incluyen indicaciones, gramática y un diálogo lógico (o flujo de llamada). Las indicaciones o mensajes del sistema, generalmente, son audios grabados que se reproducen al usuario durante el diálogo. La gramática establece las distintas respuestas que los usuarios pueden decir ante cada una de las indicaciones del sistema, esto es de gran importancia debido a que el sistema solo entenderá las palabras o frases que hayan sido definidas en la gramática. Finalmente, el diálogo lógico define las acciones que debe tomar el sistema [136].

#### 3.9.3 Interfaces de usuario de video

Por medio de las interfaces de usuario de video (VUI) es posible operar aplicaciones utilizando sistemas de visión, los cuales son capaces de reconocer gestos y movimientos [137], para realizar dicho reconocimiento se debe hacer un procesamiento de imágenes. En el trabajo de G. Iannizzotto *et al.* [138] se incluyen las etapas de pre-procesamiento, análisis de firmas y clasificación de gestos, para el reconocimiento de gestos y operación de los comandos establecidos para el sistema.

#### 3.10 Captura de movimiento

La captura de movimiento (mocap) se refiere al proceso de muestreo y registro de movimientos, ya sea de seres humanos, animales u objetos [139]-[141]. La información

adquirida mediante *mocap* puede ser usada para estudiar el movimiento o para dar una ilusión de vida a modelos informáticos 2D o 3D [139], [142].

Existen diversos tipos de sistemas para *mocap* dentro de los cuales se pueden mencionar los magnéticos, electromecánicos y ópticos.

## 3.10.1 Sistemas magnéticos

Los sistemas magnéticos (o electromagnéticos) normalmente utilizan de 11 a 18 sensores, una unidad de control y software. Los sensores de seguimiento son colocados en el objeto a capturar para medir la relación espacial que hay hasta el transmisor magnético. El transmisor genera una baja frecuencia electromagnética que es detectada por los receptores y una entrada en una unidad de control electrónica donde ésta es filtrada y amplificada. Una vez hecho esto, la frecuencia se manda a la computadora central en donde el software determina la posición (en coordenadas espaciales) y la orientación para cada sensor [142].

### 3.10.2 Sistemas electromecánicos

Estos sistemas se basan en estructuras relacionadas a potenciómetros, giroscopios o dispositivos similares para la medición angular, los cuales se localizan en las principales articulaciones del cuerpo. Las nuevas versiones utilizan sensores de inercia MEMS (*Microelectromechanical Systems*) que se ponen sobre un traje de lycra, con esto es posible medir los ángulos de las extremidades [142].

#### 3.10.3 Sistemas ópticos

Un sistema óptico típico se basa en una sola computadora que controla entre 8 y 32 cámaras digitales CCD (*Charge-Coupled Device*) [142]. En la mayoría de este tipo de sistemas las cámaras capturan la posición de marcas reflectantes (pasivas) o luminiscentes (activas). Las marcas pasivas están hechas de materiales reflectantes y sus formas son esféricas, semiesféricas o circulares. A diferencia de las marcas pasivas, las marcas activas son LEDs (*Light-Emitting*)

*Diodes*). Algunos de estos sistemas de marcas activas iluminan un LED a la vez, evitando la necesidad de identificar cada marca [139].

## 3.10.4 Sensores RGBD

Una alternativa a los sistemas ópticos compuestos por cámaras CCD son las cámaras de profundidad, las cuales se han empleado durante muchos años en el área de visión por computadora. Inicialmente las cámaras de profundidad que existían eran altamente costosas y de baja calidad, pero con la aparición de sensores de profundidad de bajo costo como Kinect<sup>®</sup> [143] estos inconvenientes quedaron atrás [144]. Los sensores de profundidad o sensores RGBD adquieren información de color (RGB) y profundidad, ésta última se obtiene por medio de un sensor infrarrojo [145], [146]. Para visualizar la profundidad, los sensores RGBD generan un mapa de profundidad en donde los objetos se identifican con un nivel de gris dependiendo de su cercanía al sensor, los elementos más distantes tienen un nivel de gris bajo mientras que los cercanos poseen un nivel de gris alto (Figura 3.14).



Figura 3.14. Mapa de profundidad de un sensor RGBD.

Una característica importante de los sensores RGBD es su campo de visión, ya que éste define la cantidad de elementos que puede capturar el sensor (Figura 3.15).



Figura 3.15. Campo de visión de un sensor RGBD.

Los sensores de profundidad suelen usarse para conseguir información espacial del seguimiento de una persona, a partir de la información de profundidad capturada por el sensor infrarrojo del sensor RGBD [146]. Para poder procesar la información adquirida por los sensores, cada fabricante desarrolla un SDK. Actualmente existen diversos modelos de sensores RGBD, algunos de los más utilizados son Kinect<sup>®</sup> en su versión 1 y 2, PrimeSense<sup>®</sup>, RealSense<sup>®</sup> [147], Asus Xtion Pro<sup>®</sup> [148] y PointGrab<sup>®</sup> [149], entre otros [150]-[152].

## 3.11 Interfaces de programación de aplicaciones

Las interfaces de programación de aplicaciones (APIs) permiten a terceros y desarrolladores el acceso a datos y servicios para construir aplicaciones de manera rápida, esto se debe a que las APIs promueven el reúso de código. El reúso de código se puede hacer de dos maneras, copiando el código del API y colocándolo en el nuevo archivo de código o llamando a funciones que hayan sido empaquetadas y destinadas explícitamente para ser reutilizadas [153], [154].

Mediante una API, es posible que dos aplicaciones computacionales puedan comunicarse a través de una red utilizando un lenguaje en común. Una API es esencialmente un convenio que establece la conexión entre el proveedor y el consumidor de manera eficiente ya que las interfaces son consistentes, predecibles y están documentadas [153].

Existen APIs que se encuentran disponibles para cualquier desarrollador, otras están abiertas solo para socios y algunas otras solo son utilizadas internamente para facilitar el trabajo entre equipos. Algunas de las APIs más reconocidas son Facebook<sup>®</sup> [155], Twitter<sup>®</sup> [156], Google Maps<sup>®</sup> [157] y Netflix<sup>®</sup>, entre otras.

#### 3.12 Aplicaciones multimedia

De acuerdo a A. Durresi y R. Jain [158] las aplicaciones multimedia pueden clasificarse como se muestra en la Figura 3.16.



Figura 3.16. Clasificación de las aplicaciones multimedia.

Muchas aplicaciones multimedia son altamente sensibles al retraso *end-to-end* y a la fluctuación, pero pueden tolerar la pérdida de información. En las aplicaciones de tiempo real, como voz a través de internet y videoconferencias, la información no puede ser almacenada por mucho tiempo, ya que se debe suavizar el retraso de la red y la fluctuación. En las aplicaciones no interactivas de transmisión de audio/video almacenado, el audio y video se guardan en un servidor y posteriormente el usuario comienza a reproducir los archivos mientras que éste se recibe, una vez iniciada la reproducción ésta debería avanzar de acuerdo al tiempo de grabación original [158].

## **IV. PROTOCOLOS DE RED DE TIEMPO REAL**

Los protocolos son conjuntos de reglas que deben seguirse durante el curso de alguna actividad, si no se sigue el protocolo la actividad no tendrá éxito. En los protocolos de comunicación, el conjunto de reglas rige el intercambio de información, y la comunicación falla si éstas no se siguen correctamente. La comunicación entre computadoras está basada en el intercambio de datos en donde se transfiere un mensaje, como en el diagrama de secuencia de la Figura 4.1. Dependiendo del sistema, un mensaje puede ser una simple señal eléctrica o una gran cantidad de datos [159].



Figura 4.1. Intercambio de mensajes en un protocolo simple.

En los protocolos de comunicación el intercambio de datos puede darse entre dos o más partes (*peers*), permitiendo que haya hasta *N-peers* dentro de dicho intercambio. Para cada uno de los *N-peers* en una comunicación *N-peer*, el protocolo define un lenguaje cuyas sentencias son las secuencias de los mensajes recibidos por dicha parte y cuyo alfabeto de símbolos es el conjunto de todos los mensajes posibles [159].

#### 4.1 Arquitecturas para la comunicación de redes

Uno de los principios de diseño para redes es el uso de capas. En la comunicación de redes regularmente se utiliza este tipo de estructura o arquitectura, con la cual es posible dividir las tareas de comunicación en partes más pequeñas para cumplir con sub-tareas [160]. Existen principalmente dos tipos de arquitecturas basadas en el esquema de capas, estas arquitecturas son la OSI (véase 4.1.1) y la TCP/IP (véase 4.1.2), no obstante, también existen algunas más recientes como la que se utiliza para IoT (véase 4.1.3).

## 4.1.1 Modelo OSI

El modelo OSI (*Open Systems Interconnection*) fue desarrollado en 1984 por la ISO (Organización Internacional de Normalización) como un marco de referencia de estándares para la comunicación de redes. El modelo OSI ha sido considerado como el principal modelo de comunicación entre computadoras y redes [160]. Este modelo define el proceso de comunicación en 7 capas, tal y como se muestra en la Figura 4.2.



Figura 4.2. Modelo OSI de 7 capas.

Cada capa está debidamente autocontenida, lo que permite que las tareas asignadas a cada capa puedan llevarse a cabo de manera independiente [160]. Una capa no define un solo protocolo sino una función de comunicación de datos que puede realizarse por diversos protocolos [161]. En este tipo de modelos de comunicación los datos se manejan de maneras distintas dependiendo de la capa. La unidad en la cual una determinada capa manipula la información se denomina PDU (*Protocol Data Unit*) [162].

## 4.1.1.1 Capa física

Esta capa define las características del hardware necesario para la transmisión de datos, por ejemplo, el nivel de voltaje o el número y la ubicación de los pines de la interfaz [161]. También se encarga de la trasmisión física de las señales eléctricas (o electromagnéticas) entre los diferentes sistemas [163]. Además, esta capa es la responsable de convertir los datos digitales en un flujo de bits para su transmisión en internet [164].

#### 4.1.1.2 Capa de enlace de datos

El estándar IEEE 802.2 define el enlace de datos como un ensamble de dos o más instalaciones de una terminal y la interconexión de canales de comunicación que operan de acuerdo a un método particular, permitiendo el intercambio de información [165]. En este mismo estándar se definen las especificaciones para el control del enlace lógico.

En la capa de enlace de datos se realiza un servicio similar al de la capa física [163]. Esta capa representa el enlace básico de comunicación que existe entre las computadoras y es responsable de enviar *frames* o paquetes de datos sin errores [164]. En esta capa se llevan a cabo principalmente dos tareas, la detección de errores y el control de flujo.

#### 4.1.1.3 Capa de red

La capa de red es responsable de administrar las conexiones a través de la red así como la transmisión de datos [161], [164].

El modelo OSI utiliza mayormente redes de conmutación de paquetes debido a que son consideradas el tipo de red más eficiente para transmitir datos desde diferentes puntos de vista. En la capa de red existen estaciones terminales y nodos de conmutación [163].

## 4.1.1.4 Capa de transporte

Esta capa permite que se tenga una conexión fiable sobre cualquier tipo de red, su función principal es asegurar la calidad de transmisión [163]. La capa de transporte es responsable de asegurar que los mensajes se entreguen en la secuencia correcta y sin errores. Esta capa divide los mensajes en segmentos más pequeños y provee un control de tráfico de red de mensajes. El control de tráfico es una técnica que se utiliza para asegurar que una fuente no abrume con datos un destino, cuando se reciben estos datos se ocupa una cierta cantidad de procesamiento antes de que el *buffer* esté libre y listo para recibir más datos [164].

#### 4.1.1.5 Capa de sesión

La capa de sesión administra las sesiones (o conexiones) entre las aplicaciones [161], de igual manera gestiona la recuperación de caídas de red y los protocolos de sincronía entre aplicaciones [163]. Esta capa maneja los datos en la manera en que se reciben sin dividirlos o concatenarlos [162] y es responsable de reportar las excepciones de una sesión así como de coordinar los modos de envío y recepción en un intercambio de datos [164].

## 4.1.1.6 Capa de presentación

Esta capa es responsable de la manera en la que se presentan los datos a la aplicación. Al inicio de la comunicación la capa de presentación negocia la forma de los datos que se van a transferir a otra entidad (sintaxis de transferencia). Después de esta negociación la capa de transporte puede proveer servicios adicionales como compresión, encriptación y traslación, esto dependiendo de la aplicación [162]. Gracias a esta negociación diferentes plataformas se pueden entender al conectarse a través de una misma red, ya que se define una manera universal para codificar la información [163].

#### 4.1.1.7 Capa de aplicación

En la capa de aplicación se localizan los programas, dentro de los cuales se pueden encontrar servidores y clientes, entre otros [163]. La capa de aplicación es el nivel más alto del modelo OSI y es responsable de proveer aplicaciones para el usuario final así como funciones para administrar la red [164]. Los protocolos de esta capa varían de acuerdo al tipo específico de dato que el usuario quiere transferir. Esta capa también define los parámetros aceptables QoS (*Quality of Service*) para cada servicio [162].

#### 4.1.2 Modelo TCP/IP

El nombre de TCP/IP se refiere a todo un conjunto de protocolos de comunicación de datos. Dicho conjunto adquiere su nombre de dos de los protocolos que pertenecen a éste, los cuales son el TCP (Protocolo de Control de Transmisión) y el IP (Protocolo de Internet). El protocolo TCP/IP se conoce también como IPS (Conjunto de Protocolos de Internet). TCP/IP crea una red heterogénea con protocolos abiertos e independientes al sistema operativo. Los protocolos TCP/IP están disponibles para todos y son desarrollados y modificados por consenso y no por un solo fabricante [161].

La arquitectura TCP/IP se compone de menos capas que el modelo OSI. Así mismo, existen diferentes descripciones del modelo TCP/IP en donde el número de capas va desde 3 a 5. En la Figura 4.3 se muestra el modelo de 4 capas.



Figura 4.3. Arquitectura TCP/IP de 4 capas.

#### 4.1.2.1 Capa de acceso a red

Los protocolos de esta capa proveen los medios al sistema para entregar información a otros dispositivos en una red directamente adjunta. Esta capa define cómo utilizar la red para transmitir un datagrama<sup>2</sup> IP. Existen varios RFCs que describen cómo se maneja IP y las interfaces con protocolos existentes como Ethernet, Token Ring, FDDI, HSSI, PPP, X.25, Frame Replay y ATM [160], [161], [164].

## 4.1.2.2 Capa de internet

En cuanto a la capa de red o internet, el protocolo más importante es IP, sin embargo, como se muestra en la Tabla 4.1, existen más protocolos para esta capa [166].

Protocolo	Nombre	Función
IP	Internet Protocol.	Enruta los paquetes del emisor al
		receptor.
ICMP	Internet Control Message Protocol.	Maneja la información respecto al
		enrutamiento basado en IP y el
		comportamiento de la red.
PING	Packet Internetwork Groper.	Verifica la accesibilidad y el tiempo
		de envío y respuesta entre un emisor
		y receptor de un par de direcciones
		IP.
ARP	Address Resolution Protocol.	Convierte las direcciones de red IP y
		MAC en un segmento de cable
		específico.
RARP	Reverse Address Resolution Protocol.	Convierte una capa de direcciones
		MAC en una dirección IP numérica.

Tabla 4.1. Protocolos para la capa de internet del modelo TCP/IP de 4 capas.

<sup>&</sup>lt;sup>2</sup> Un datagrama es un mensaje corto del programa del cliente al programa del servidor. En un flujo de *socket* un mensaje consiste de un número de paquetes IP, los cuales se envían a través del flujo *socket*.

воотр	Bootstrap Protocol.	Gestiona la asignación de direcciones IP a la red.
RIP	Routing Information Protocol.	Define el vector original de distancia y el protocolo más básico de enrutamiento.
OSPF	Open Shortest Path First.	Define un protocolo de enrutamiento de estado de enlace.
BGP	Border Gateway Protocol.	Define un protocolo de enrutamiento que se conecta a las comunes redes troncales de internet.

# 4.1.2.3 Capa de transporte

La capa de transporte provee servicios para la entrega de información *end-to-end*. Los dos protocolos más importantes para la capa de transporte son TCP (véase 4.3.1) y UDP (véase 4.3.2) [160], [161].

# 4.1.2.4 Capa de aplicación

En la capa de aplicación se encuentran las aplicaciones y procesos que utiliza la red. Esta capa incluye todos los procesos que usan los protocolos de la capa de transporte para entregar la información [161]. Los principales protocolos empleados en esta capa se muestran en la Tabla 4.2.

Protocolo	Nombre	Función
Telnet	Network Terminal Protocol.	Brinda un acceso remoto sobre la red.
FTP	File Transfer Protocol.	Utilizado para transferencia interactiva de archivos.
SMTP	Simple Mail Transfer Protocol.	Realiza la entrega de correo electrónico.

Tabla 4.2. Protocolos	para la capa	de aplicación de	el modelo TCP/IF	de 4 capas.
14014 1121 11010000100	para la capa	at aprilation a	•••••••••••••	at . tapas.

HTTP	Hypertext Transfer Protocol.	Lleva a cabo la entrega de páginas web sobre la red.
DNS	Domain Name System.	Mapea direcciones IP a los nombres asignados a los dispositivos de red.
OSPF	Open Shortest Path First.	Usado por los dispositivos de red para intercambiar información de ruteo.
NFS	Network File System.	Permite que los archivos sean compartidos en la red por varios alojamientos.

## 4.1.3 Modelo para IoT

El término internet de las cosas (IoT, *Internet of Things*) fue propuesto en 1999 y se refiere a la construcción de una red, basada en internet, que conecta todas las cosas en el mundo mediante el uso de tecnologías relacionadas, para realizar la identificación automática de las cosas y el intercambio de información. En la actualidad, el internet de las cosas ha sido redefinido con base en las diferentes perspectivas y escenarios de aplicación [167]. Existen diferentes tipos de arquitecturas propuestas para IoT, algunas de las cuales parten de modelos basados en capas. Para obtener una representación básica de IoT se utiliza un modelo de 3 capas (Figura 4.4) que se compone de los niveles de percepción, red o internet y aplicación-interface [167]-[169].



Figura 4.4. Arquitectura básica de 3 capas para IoT.

La capa de percepción consiste de varios sensores y actuadores y su función es identificar y controlar cosas, así como también reunir información. Por su parte, la capa de red o internet incluye una variedad de redes, las cuales pueden ser privadas, móviles, redes de área local (LAN) y redes de área amplia (WAN). Finalmente, la capa de aplicación es la interface de los servicios de IoT y los usuarios [167].

Otros tipos de arquitecturas que se pueden encontrar en la literatura son las arquitecturas orientadas a servicios (SoA) [170], [171]. Una arquitectura SoA genérica consta de cuatro capas (Figura 4.5), las cuales son percepción, red o internet, servicio y aplicación-interface. A diferencia de la arquitectura básica de 3 capas, la arquitectura SoA consta de una capa de servicio que provee y administra los servicios requeridos por los usuarios y aplicaciones [171].

Existen proyectos como IoT-A (*Internet of Things - Architecture*) que han tratado de diseñar una arquitectura común basada en las necesidades de investigadores y la industria [169], [172], [173], ésta se muestra en la Figura 4.6. Cabe mencionar que existe una diversidad de arquitecturas para IoT [168], [174], [175], esto dependiendo de los sistemas y su área de implementación.



Figura 4.5. Arquitectura SoA genérica.



#### 4.2 Servicios web

El concepto de servicio web está estrechamente relacionado con el término protocolo ya que es un elemento esencial para su descripción, validación y uso. Un servicio web es una interfaz que permite el acceso a las funciones de un objeto independientemente de las tecnologías o plataformas utilizadas [128].

Las redes computacionales brindan servicios web que permiten el intercambio de datos, programas, archivos multimedia u otros documentos. Los servicios web pueden ser usados para la descarga de programas, inicio remoto, actualización remota de bases de datos o para acceder a sitios web. Así mismo, los servicios web ayudan a los usuarios a comunicarse con otros [176].

Las características de los servicios web se pueden derivar en dos clases, la primera está asociada con el tipo de suposiciones lógicas que se pueden hacer con base en el tipo de información que se va a transferir. La segunda clase está relacionada con los aspectos técnicos concernientes al costo o a la capacidad del sistema de comunicación [159].

A continuación se presentan algunas de las características más comunes de los servicios web.

## 4.2.1 Secuencia de preservación

La secuencia de preservación es una propiedad del proveedor de servicios web de entregar mensajes en el mismo orden en el que fueron enviados [159].

#### 4.2.2 Sincronización de unidades de datos

Esta característica se refiere a la correspondencia uno a uno que existe entre los mensajes que pasan al servicio web para ser transmitidos y los mensajes entregados a los receptores. A este tipo de servicios web se les conoce como servicios web orientados a mensaje o servicios web orientados a bloques ya que entregan bloques de datos en su totalidad.

Otro tipo de servicios web son los orientados a la transmisión. En estos, los límites que existen entre las unidades de datos suministradas al servicio web no necesariamente se conservan cuando la información se entrega al receptor. Los datos se consideran como parte de un flujo que puede cortarse en pedazos y entregarse en unidades de un tamaño conveniente [159], esto se muestra en la Figura 4.7.



Figura 4.7. Servicio orientado a bloques (imagen superior), servicio orientado a transmisión (imagen inferior).

# 4.2.3 Control de flujo

El control de flujo asegura que los mensajes no se entreguen más rápido de lo que el receptor pueda manejar. Este principio también se aplica a la sincronización realizada a través de un *buffer* finito [159].

# 4.2.4 Libre de error

Un servicio web libre de errores entrega los mismos mensajes que fueron enviados sin ningún tipo de falla [159]. El servicio web puede darse bajo los siguientes cuatro posibles escenarios:

- Sin fallas.
- Falla de pérdida de mensajes.
- Falla de mensajes falsos.
- Falla de mensajes dañados.

Cualquier otro tipo de fallas se pueden expresar como una combinación de las cuatro anteriores.

#### 4.2.5 Reinicio del servicio

Idealmente los servicios web siempre se encuentran operando sin detenerse o reiniciarse, sin embargo, para fines prácticos es necesario establecer un estado inicial estándar del cual pueda partir el servicio web en caso de requerir un reinicio. Las características de un reinicio regularmente dan lugar a la pérdida de una cantidad arbitraria de datos [159].

#### 4.2.6 Establecimiento y liberación de la conexión

Muchos servicios web requieren que los usuarios establezcan una conexión mediante la configuración de un canal lógico o físico antes de que se lleve a cabo el intercambio de datos [159]. La conexión entre usuarios se da en 3 fases principales: 1) establecimiento de conexión, 2) transferencia de datos y 3) liberación de conexión, tal y como se muestra en la Figura 4.8.



Figura 4.8. Fases de operación de una conexión.

La primer fase es la de establecimiento de conexión, en ésta se establece un canal por medio del intercambio de un tipo particular de mensajes, el cual permite a los usuarios del servicio web establecer su estado global inicial. Esta fase puede utilizarse para acordar alguna clase de parámetros que describan la fase de transferencia de datos. De esta manera, los usuarios del servicio web ya no tienen que enviar todos los parámetros en cada mensaje, haciendo que el servicio web sea más eficiente ya que se evita la transmisión repetitiva de información. Una vez que se acuerdan los valores para los parámetros se inicia la fase de transferencia de datos en donde se lleva a cabo el intercambio de información de manera subsecuente. La transferencia de datos se realiza hasta que se interrumpe la conexión y se libera el canal en la fase de liberación de conexión [159].

# 4.2.7 Cambio de modo

En el diseño de protocolos existen diversos modos de operación de un servicio web, dentro de los cuales se pueden mencionar los siguientes:

- Orientado a conexión
- Sin conexión.
- Punto a punto.
- Multipunto.
- Simplex.
- Half-duplex.
- Full-duplex.

# 4.2.7.1 Modo de operación orientado a conexión

La comunicación orientada a conexión necesita que se haya establecido una conexión antes de comenzar la transmisión de datos del emisor al receptor [162]. Un ejemplo de este modo de conexión es la comunicación por teléfono, ya que no se puede iniciar una conversación hasta que se haya establecido correctamente la conexión y el usuario receptor atienda la llamada.

# 4.2.7.2 Modo de operación sin conexión

La comunicación sin conexión se refiere a aquella en donde no es necesario establecer una conexión antes de realizar la transmisión de datos. El control de la información se añade a los datos y estos se envían a su destino, sin poder conocer con precisión si se recibió o no la información [162]. Como ejemplo se puede considerar el envío de un correo, todo lo que puede hacer el emisor es escribir la dirección de destino en el mensaje y entregarlo en la oficina de correo, sin poder conocer con certeza si el destinatario recibirá apropiadamente el mensaje.

# 4.2.7.3 Modo de operación punto a punto

En este tipo de operación dos servicios web de usuario se comunican uno con el otro, esto se hace por medio de un canal físico o lógico que los interconecta [159].

## 4.2.7.4 Modo de operación multipunto

A diferencia del modo punto a punto, el modo multipunto permite que múltiples usuarios se comuniquen con otro. Los servicios web del modo multipunto pueden denominarse *broadcast* o *multicast*. En el caso del *broadcast* todos los usuarios disponibles pueden recibir un mensaje enviado por uno de ellos, mientras que en el *multicast* el emisor puede seleccionar un subconjunto de usuarios para que reciban un mensaje o mensajes determinados. También puede mencionarse el servicio web *inverse broadcast*, en el cual un solo receptor es capaz de recibir simultáneamente los mensajes enviados por los demás usuarios [159].

# 4.2.7.5 Modo de operación simplex

En este modo, la comunicación involucra la transmisión de datos en una dirección todo el tiempo. Como se muestra en la Figura 4.9, la información fluye solamente en un sentido [162].



Figura 4.9. Modo de operación simplex.

En la Figura 4.10 se muestra un ejemplo básico del modo de operación simplex.



Figura 4.10. Flujo de información de una estación de radio a un radio.

En este caso se muestra el flujo de información de una estación de radio, la cual transmite información a través de una antena para ser captada por un radio. Dado que el flujo de datos se da en una sola dirección, el radio puede captar la información proveniente de la antena pero no puede enviar datos a la estación de radio.

### 4.2.7.6 Modo de operación half-duplex

En este modo de operación, dos *peers* comparten el mismo canal de comunicación para enviar y recibir información en una base de tiempo compartido. Cuando un *peer* A envía un dato a un *peer* B el canal de comunicación se ocupa, por lo que el *peer* B no puede enviar datos hasta que el canal de comunicación se haya liberado (Figura 4.11). Esto quiere decir que el flujo de datos se lleva a cabo en una sola dirección en un momento determinado [162].



Figura 4.11. Modo de operación half-duplex.

Un ejemplo de este modo de operación se muestra en la Figura 4.12, en donde se describe la transmisión de información entre dos radios portátiles (*walkie-talkie*). El funcionamiento de estos dispositivos implica que un usuario puede comunicarse con otro mientras el canal este libre, esto es, cuando un usuario presiona el botón para hablar, el canal de comunicación se ocupa dejando al otro usuario solo como oyente del mensaje, una vez que se deja de presionar el botón el canal se libera permitiendo que otras personas puedan hacer uso de él.



Figura 4.12. Flujo de información entre dos radios portátiles (walkie-talkie).
## 4.2.7.7 Modo de operación full-duplex

El modo *full-duplex* utiliza dos canales de comunicación para la transmisión y recepción de datos, gracias a ello es posible tener un flujo de información en ambos sentidos (Figura 4.13) al mismo tiempo [162].



Figura 4.13. Modo de operación full-duplex.

Un claro ejemplo de este modo de operación es la comunicación por teléfono. Como se observa en la Figura 4.14, dos usuarios pueden intercambiar información de manera simultánea sin necesidad de esperar a que el canal de comunicación esté libre, ya que en el modo *full-duplex* existen dos canales.



Figura 4.14. Flujo de información en la comunicación vía teléfono.

## 4.3 Protocolos de comunicación

Tomando como base las arquitecturas descritas en la sección 4.1, a continuación se presentan algunos de los protocolos más utilizados por estos modelos, especialmente para la transmisión de datos y procesos relativos a ello.

# 4.3.1 IP

El internet basado en IP realiza la entrega de paquetes sin garantías por lo que estos pueden perderse o entregarse en desorden, así mismo el retraso es impredecible. La ventaja de este tipo de modelos es que mantiene a los enrutadores tan simples como es posible y hace que la arquitectura de la red sea muy escalable [158].

El administrador de red asigna una dirección IP única a cada anfitrión en una red [177].

## 4.3.2 TCP

El TCP tiene un control de velocidad que permite que el tráfico se adapte por sí mismo de acuerdo a las condiciones de red [178].

Para incrementar la confianza en los servicios *end-to-end*, los paquetes de datos se transportan mediante el TCP, la cual es una de las capas de transporte más utilizadas en redes IP [158].

El TCP tiene mecanismos para encubrir la falta de confiabilidad del IP y presenta un canal de red confiable para aplicaciones. Por ejemplo, el TCP retransmite los paquetes perdidos y los mantiene en orden, también tiene mecanismos para mantener bajo control la congestión de red y para evitar el colapso de congestión. Es por ello que el TCP es una solución óptima para el tráfico de datos [158].

#### 4.3.3 UDP

El UDP (*User Datagram Protocol*) es otra de las capas de transporte del IP. Es más ligero que el TCP por lo cual es una mejor opción para utilizarse en aplicaciones de tiempo real, sin embargo incrementa los requerimientos de ancho de banda pudiendo ocasionar un desbordamiento en las redes con una velocidad descontrolada del tráfico UDP [178]. Este descontrol podría generar problemas de congestión ya que el UDP no tiene control sobre esto [179].

## 4.3.4 RTP

El RTP (*Real-time Transport Protocol*) es un protocolo de red diseñado específicamente para aplicaciones de tiempo real, especialmente en aplicaciones multimedia de transmisión de audio

y video (telefonía, video conferencias y telepresencia) donde se requiere un control de velocidad de transmisión. El RTP no tiene un control de velocidad funcionalmente implementado en la especificación, sin embargo sí se encarga de adquirir información del estado de la red a través del estado de los paquetes RTCP. El RTP trabaja en conjunto con el RTCP, el cual entrega paquetes de reportes periódicos a ambas partes de la comunicación RTP sobre el estado de la red [178], [179].

En la mayoría de las implementaciones de VoIP (*Voice over IP*) el RTP es utilizado como una capa de aplicación de protocolo para el intercambio de datos multimedia, la cual se ejecuta sobre el UDP ya que requiere la entrega oportuna de información y puede tolerar la pérdida de algunos paquetes. Una sesión RTP se caracteriza por tener un par de direcciones IP y puertos, además de la codificación de audio utilizada (tipo de carga útil), así como la fuente de la transmisión de paquetes RTP (SSRC), el espacio de *timestamp* y el espacio de secuencia [180].

El TCP tiene cuidado de retransmitir los paquetes perdidos, sin embargo no asegura el retraso ni su variación. Es por ello que el tráfico multimedia no trabaja bien con el TCP [158].

#### 4.4 WebRTC

WebRTC es un *framework* de libre uso para la web que habilita las comunicaciones de tiempo real (RTC) en los navegadores, facilitando entre ellos el intercambio multimedia en tiempo real con un modo de operación punto a punto. El WebRTC está siendo estandarizado como API por el W3C con la finalidad de definirlo como un API que habilite una aplicación web que pueda operar en cualquier dispositivo a través de un acceso seguro a los periféricos de entrada (cámaras y micrófonos) [181]. Por su parte el IETF (*Internet Engineering Task Force*) está estandarizando el WebRTC como protocolo [182].

El WebRTC se ha enfocado principalmente en aplicaciones como videoconferencias y la transmisión de audio a través de internet [183], no obstante en la actualidad se han estado explorando otras áreas de oportunidad para utilizar el WebRTC impulsando la implementación de diversos sistemas con base en este *framework*.

### 4.4.1 Ventajas del uso de WebRTC

Una de las ventajas del WebRTC es que es se puede acceder a cada uno de sus componentes por medio de una API de JavaScript, permitiendo que los desarrolladores implementen fácilmente sus propias aplicaciones web utilizando las RTC [182].

### 4.4.2 Arquitectura web

La semántica de arquitectura web clásica está basada en un paradigma cliente-servidor [181]. En la Figura 4.15 se puede apreciar el modelo básico de una aplicación web. En este esquema la información entre el navegador y el servidor web se transporta por medio del HTTP (*Hyper-Text Transport Protocol*), el cual corre sobre el TCP o en algunas nuevas implementaciones sobre el protocolo WebSocket. La aplicación está implementada en HTML (*Hyper-Text Markup Language*) que por lo regular incluye JavaScript y CSS (*Cascading Style Sheets*) [184].



Figura 4.15. Modelo de un navegador web.

En un caso simple el navegador envía una solicitud HTTP al servidor web y éste envía una respuesta con la información solicitada. En casos más complejos el servidor envía JavaScript el cual corre en el navegador interactuando con él a través de APIs y con el usuario. El navegador intercambia información con el servidor por medio de un HTTP abierto o de canales WebSocket

[184]. Los recursos otorgados por el servidor están estrechamente asociados a una entidad conocida como URI (*Uniform Resource Identifier*) o URL (*Uniform Resource Locator*) [181].

## 4.4.3 Función RTC en el navegador

En la Figura 4.16 se muestra el modelo del navegador y el rol del RTC en él. La función del RTC interactúa con la aplicación web utilizando APIs estándar, comunicándose con el sistema operativo utilizando el navegador. Una de las características del WebRTC es la interacción que se da de navegador a navegador, esto se conoce como *peer connection*. La función RTC en un navegador se comunica utilizando protocolos estándar *on-the-wire* con la función RTC en otro navegador o VoIP o aplicación de video [184].



Figura 4.16. Función RTC en el navegador web.

## 4.4.4 Escenario de conferencias WebRTC

El escenario de conferencias que utiliza WebRTC se basa en la topología de red tipo malla (Figura 4.17), esta clase de arreglo permite que cada navegador sea capaz de recibir y manejar el flujo multimedia generado por los demás navegadores, así mismo cada navegador puede enviar su propio flujo multimedia a los demás participantes de la conferencia.

Para un navegador web resulta difícil operar en este tipo de escenario ya que debe manejar el flujo multimedia y al mismo tiempo incrementar linealmente la disponibilidad de ancho de banda de la red. Por ello, en los sistemas de videoconferencias se utiliza una topología de red tipo estrella en donde cada navegador se conecta a un servidor dedicado [181], ver Figura 4.18.





Figura 4.17. Topología de red tipo malla.

Figura 4.18. Topología de red tipo estrella.

El servidor se encarga de controlar los recursos de la conferencia y de negociar los parámetros con cada uno de los pares de la red, además agrega o mezcla cada uno de los flujos multimedia y también distribuye apropiadamente la mezcla de flujos a cada uno de los participantes de la conferencia. Este servidor puede ser uno de los mismos pares de la red o bien un servidor optimizado específicamente para procesar y distribuir información en tiempo real, a este último se le conoce como unidad de control multipunto (MCU) [181].

#### 4.4.5 Proceso de comunicación WebRTC

En la Fig. 4.19 se muestra el proceso general para comunicar dos navegadores web utilizando WebRTC para establecer una videollamada simple entre ellos [185].



Figura 4.19. Proceso general de comunicación WebRTC.

A continuación se describen cada una de las etapas del proceso de comunicación WebRTC.

- **Conectar usuarios:** La forma más simple para conectar a los usuarios es que visiten un mismo sitio web que pueda identificar cada navegador mediante un *token* único y pueda conectarlos mediante un servidor de señalización compartido.
- Iniciar las señales: Una vez que los usuarios han compartido un *token* pueden comenzar a intercambiar mensajes de señalización para negociar la configuración de la conexión WebRTC.
- Encontrar los candidatos: En esta etapa los navegadores intercambian información sobre sus redes y su forma de conexión.
- **Negociar las sesiones multimedia:** En la negociación se define el tipo y formato multimedia que se utilizará durante el intercambio de información.
- Iniciar las transmisiones RTCPeerConnection: Una vez que todas las etapas anteriores han sido completadas los navegadores están listos para comenzar a realizar la transmisión multimedia.

# 4.4.6 Arquitectura WebRTC

El WebRTC extiende la semántica cliente-servidor introduciendo un paradigma de comunicación punto a punto entre navegadores [181]. En la Figura 4.20 se muestra un conjunto de elementos típicos de un sistema WebRTC incluyendo servidores web corriendo en distintos sistemas operativos y en diversos tipos de dispositivos. Otros elementos que se pueden incluir son puertas de enlace al PSTN (*Public Switched Telephone Network*) y otros nodos de comunicación de internet como teléfonos SIP (*Session Initiation Protocol*) y clientes Jingle. El WebRTC habilita la comunicación entre todos los dispositivos [184].



## 4.4.6.1 Triángulo WebRTC

El escenario WebRTC más común es donde ambos navegadores ejecutan la misma aplicación web descargada de la misma página web [181], con esto se da lugar al triángulo WebRTC que se puede apreciar en la Figura 4.21.



Figura 4.21. Triángulo WebRTC.

#### 4.4.6.2 Trapezoide WebRTC

En el trapezoide WebRTC los navegadores ejecutan una aplicación web descargada de diferentes servidores web los cuales se comunican mediante un protocolo estándar de señalización como SIP. Los mensajes se transportan a través de los protocolos HTTP o WebSocket por medio de servidores web que pueden modificarlos, traducirlos o administrarlos según sea necesario [181], [184], ver Figura 4.22.

La señalización entre el navegador y el servidor no está estandarizado en el WebRTC ya que se considera que es parte de la aplicación. En casos complejos la información multimedia podría no fluir directamente entre dos navegadores pero si hacerlo a través de *relays* multimedia [184].



Figura 4.22. Trapezoide WebRTC.

#### 4.5 Pruebas de software

En la ingeniería de software existe un área dedicada a las pruebas de software (*software testing*), éstas son de vital importancia para evitar fallas que puedan ser visibles a los usuarios [186]. El área de pruebas de software está íntimamente relacionada con la calidad de software, en la Figura 4.23 se muestra el modelo de calidad para software y sistemas, propuesto en el estándar internacional ISO 9126 [187].



Figura 4.23. Modelo de calidad interna y externa según el ISO 9126 [187].

Como se puede apreciar, el modelo se conforma de 6 principales características, las cuales poseen a su vez subcaracterísticas. Dichas características pueden ser clasificadas en dos tipos, características funcionales y características no funcionales. Las características funcionales incluyen las funciones y funcionalidad que provee el sistema, mientras que las no funcionales comprenden otras características como confiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad. Cada una de estas características (funcionales y no funcionales) puede evaluarse utilizando distintas técnicas de prueba [186].

Las pruebas de software se pueden dividir principalmente en 3 tipos, los cuales son: caja blanca, caja negra y caja gris. Las pruebas de caja blanca se basan en las estructuras de código y la implementación del software, este tipo de pruebas requiere de buenas habilidades de programación [188]. Las pruebas de caja negra solamente se basan en los requerimientos y especificaciones, no requieren de conocimiento sobre la estructura o implementación del software [189], [188]. Finalmente, las pruebas de caja gris son una combinación de las dos anteriores. En éstas se hace un análisis de la estructura del software, sin embargo el análisis no es tan profundo como en las pruebas de caja blanca, ya que lo único que se requiere es entender cómo se ha implementado el software. Con este conocimiento adquirido se pueden hacer de manera más efectiva las pruebas de caja negra [188].

Además de los tipos mencionados anteriormente, existen otros tipos de pruebas [186], en la Figura 4.24 se muestran algunos de los principales.



Figura 4.24. Principales tipos de prueba.

Para este trabajo de tesis se hicieron pruebas de rendimiento, estrés y carga. Las pruebas de estrés determinan la robustez del software, la idea principal de estas pruebas es probar el sistema con una carga que cause un alojamiento de recursos en cantidades máximas [190]. Por otro lado, las pruebas de carga consisten en poner una carga en el sistema y medir su respuesta [191]. Las pruebas de rendimiento cubren un amplio rango de procedimientos de pruebas que se realizan para validar las características medibles de rendimiento [192].

# V. PROTOCOLO DE TIEMPO REAL PARA LA FUSIÓN DE SENSORES RGBD

En este capítulo se presenta el protocolo de red de tiempo real que se implementó para dar soporte a la plataforma FSA-MD publicada en [6]. Primeramente se describe de manera general en qué consiste la plataforma FSA-MD y después se mencionan cada uno de los módulos que se utilizaron para el protocolo de tiempo real.

## 5.1 Plataforma FSA-MD

La plataforma FSA-MD se diseñó a partir del trabajo desarrollado previamente en el Laboratorio de Aprendizaje Móvil del I. T. de Chihuahua [73]. Como se muestra en la Figura 2.12, el sistema inicial consistió de un módulo de captura de movimiento y un módulo de reproducción de animación de marionetas digitales. Para la captura de movimiento se utilizó un sensor RGBD PrimeSense<sup>®</sup> Carmine 1.08 (Figura 5.1), en [193] se encuentran las características de este sensor.



Figura 5.1. Sensor PrimeSense® Carmine 1.08.

Los módulos de captura y reproducción de animación de marionetas, se implementaron en el lenguaje Processing 2.2.1.

La plataforma FSA-MD (Figura 5.2) se compone de 3 partes principales: a) módulos de captura de movimiento, b) un servidor para la fusión de datos de sensores RGBD y c) reproductores de animación de marionetas digitales.



Figura 5.2. Plataforma FSA-MD.

A continuación se describen cada uno de los componentes de la plataforma FSA-MD.

### 5.2 Módulos de captura

En el sistema inicial solo existía un módulo de captura de movimiento para un solo sensor RGBD, en cambio, para la plataforma FSA-MD se buscó tener más de un módulo de captura para múltiples sensores RGBD (un sensor por cada módulo).

De manera general, la función de los módulos de captura es procesar la información adquirida por cada sensor RGBD y generar la animación de una marioneta digital. Dado lo anterior, el funcionamiento de los módulos de captura involucra 3 etapas principales: adquisición, procesamiento y visualización.

## 5.2.1 Etapa de adquisición

En la etapa de adquisición se obtiene información de la captura de movimiento de una persona, para esto, es necesario contar con un sensor RGBD por cada módulo de captura. En el desarrollo de este trabajo de tesis se contó con el sensor PrimeSense<sup>®</sup> Carmine 1.08 y el sensor Intel<sup>®</sup> RealSense VF0800 (Figura 5.3), las especificaciones técnicas de este último sensor se encuentran en [194].



Figura 5.3 Sensor Intel® RealSense diseñado por Creative Technology®, modelo VF0800.

Además del sensor RGBD (o cámara 3D) se debe contar con un SDK que ayude a interpretar y manipular la información captada por la cámara por medio de funciones preestablecidas. Por lo regular los fabricantes de cada sensor proveen los SDK, sin embargo, también existen algunos desarrollados por terceros. Para el sensor PrimeSense<sup>®</sup> Carmine se utilizó el SDK de código abierto OpenNI<sup>®</sup> [195] en su versión 1.96.

#### 5.2.1.1 Instalación de PrimeSense® y OpenNI®

Para que el sensor pueda comenzar con la adquisición de información es indispensable instalar los componentes que sean necesarios para que la computadora detecte correctamente el sensor y el SDK. En este trabajo de tesis se utilizó principalmente una computadora Macintosh con el sistema operativo Yosemite, en la Tabla 5.1 se muestran las características detalladas de este equipo.

Dado que el sistema operativo con el cual se trabajó fue Mac OS, se tuvo que instalar OpenNI<sup>®</sup> antes de conectar el sensor PrimeSense<sup>®</sup> a la computadora. Para instalar OpenNI<sup>®</sup>, se requirió tener instalado previamente el gestor de paquetes para Mac OS *Homebrew*. Para mayor facilidad, la instalación de los paquetes se hizo directamente en la terminal. Una vez instalado *Hombrew*, se instalaron las librerías *openni.rb*, *sensor.rb* y *nite.rb* del repositorio *homebrew-openni* de T. Takeuchi [196].

Equipo de cómputo				
Modelo	Mac mini (Mid 2010)			
Sistema operativo	Yosemite 10.10.5			
Memoria	8 GB 1067 MHz DDR3			
Procesador	2.4 GHz Intel Core 2 Duo			
Almacenamiento	Disco SATA de 320 GB			
Gráficos	NVIDIA GeForce 320M 256 MB			

Tabla 5.1. Características de la computadora utilizada para la instalación de OpenNI®.

Esta instalación habilita el uso de los sensores PrimeSense<sup>®</sup> y Kinect<sup>®</sup> por medio de los comandos *install sensor* e *install sensor-kinect* respectivamente, sin embargo, no se pueden instalar ambas librerías de manera simultánea.

Ya que se hizo la instalación se ejecutó el programa de ejemplo *Sample-NiSimpleViewer* (Figura 5.4) para comprobar el funcionamiento del sensor.



Figura 5.4. Programa de ejemplo *Sample-NiSimpleViewer* para el sensor PrimeSense<sup>®</sup>: a) visualización de una escena, b) visualización de una persona en la escena.

En la Figura 5.5 se muestra un programa simple para el reconocimiento de una persona utilizando la cámara 3D de PrimeSense<sup>®</sup>.

Los pasos detallados para la instalación de PrimeSense<sup>®</sup> y OpenNI<sup>®</sup> en Mac OS se encuentran en el Apéndice A.



Figura 5.5. Programa de ejemplo Sample-NiUserTracker para el sensor PrimeSense®.

# 5.2.1.2 Configuración y características de OpenNI®

El SDK OpenNI<sup>®</sup> se compone de una serie de funciones que ayudan a procesar la información adquirida por medio del sensor RGBD. Como se muestra en la Figura 3.14, por medio de OpenNI<sup>®</sup> se puede obtener un mapa de profundidad de la escena que se está visualizando a través de la cámara. Una de las mayores utilidades del SDK es que cuenta con funciones que permiten identificar el cuerpo de una persona en una escena compuesta por varios objetos (Figura 5.5). Además de esto, el SDK genera un esqueleto de un determinado número de puntos, los cuales representan algunas articulaciones del cuerpo. El número de puntos puede variar dependiendo del sensor RGBD y del SDK, mientras mayor sea el número de puntos habrá más posibilidad de representar un movimiento. En la Figura 5.6 se muestra el esqueleto de 15 puntos que genera OpenNI<sup>®</sup> para el sensor PrimeSense<sup>®</sup>.

Aunque OpenNI<sup>®</sup> ofrece un esqueleto de 15 puntos, para efectos de la plataforma FSA-MD se utilizaron solamente 10 puntos (Figura 5.7).



Figura 5.6. Esqueleto de 15 puntos proporcionado por OpenNI® para el sensor PrimeSense®.



Figura 5.7. Esqueleto de 10 puntos utilizado para la plataforma FSA-MD.

Otra característica de OpenNI<sup>®</sup> es que permite la identificación de múltiples usuarios de manera simultánea. Como se aprecia en la Figura 5.8, a cada usuario se le asigna un color una vez que es reconocido por la cámara 3D, usualmente se asignan colores distintos para cada persona, sin embargo esto no es una regla. Aunque el SDK ofrece la posibilidad de trabajar con múltiples usuarios, los módulos de captura que se diseñaron tienen como especificación actual trabajar solamente con un usuario.



Figura 5.8. Múltiples usuarios identificados por el sensor PrimeSense®.

Otro factor importante que debe tomarse en cuenta es que el esqueleto del sensor se configuró para trabajar a manera de espejo, esto es, cuando el usuario se sitúa frente a la cámara 3D los movimientos se reproducen como si la persona estuviera viendo su reflejo en un espejo.

# 5.2.1.3 Integración de la librería OpenNI<sup>®</sup> en Processing

Al igual que en el sistema inicial, uno de los módulos de captura se implementó en Processing 2.2.1. Para poder emplear el SDK de OpenNI<sup>®</sup> en Processing fue necesario importar la librería *SimpleOpenNI*, esto se hizo por medio del menú para importar librerías (Figura 5.9) y el administrador de librerías (Figura 5.10). Finalmente, para utilizar la librería *SimpleOpenNI* se añadió la línea de código **import SimpleOpenNI**.\*.

Una vez que se realizaron todas las configuraciones necesarias, se ejecutó el software para la adquisición de datos de captura de movimiento. En la siguiente sección se muestra cómo funciona el software de adquisición de datos.

Processing	File	Edit	Sketch	Tools	Help		
			Run Preser Stop	nt		೫R ∂೫R	
		1	Import	t Library.			Add Library
			Show S Add Fi	Sketch F	older	жк	dxf minim net pdf serial video
							Contributed SimpleOpenNI

Figura 5.9. Opción para importar librerías en Processing 2.

•••		Library N	lanager	
Category:	All	<b>\$</b>	simpleopenni	
SimpleOpe A simple wi can use Sin	anNI by Max Rh rapper for Open npleOpenNI you	einer NI(Kinect-Libr have to insta	ary). Before you II OpenNI.	Remove

Figura 5.10. Administrador de librerías de Processing 2.

# 5.2.1.4 Proceso de adquisición mocap

Para iniciar con el proceso de adquisición el primer paso fue conectar el sensor RGBD a la computadora por medio del puerto USB, tal como se muestra en la Figura 5.11.



Figura 5.11. Conexión del sensor RGBD a la computadora.

A continuación se ejecutó el software de captura que se implementó en Processing 2.2.1 (*kinect\_skeleton\_ImageList.pde*). En la Figura 5.12 se muestra, de manera general, el proceso de adquisición de datos que se realiza al ejecutar el software.



Figura 5.12. Proceso de adquisición de datos del software de captura.

Una vez que el sensor está preparado para comenzar con la adquisición aparece un recuadro mostrando el mapa de profundidad generado por la librería (Figura 5.13), en ese momento se

puede posicionar una persona de frente al sensor esperando a que éste sea capaz de distinguir a la persona de entre todos los demás objetos de la escena.



Figura 5.13. Módulo de captura: mapa de profundidad del sensor RGBD.

Cuando el usuario se colorea de azul quiere decir que ha sido identificado correctamente (Figura 5.14), si el usuario se mueve a lo largo de la escena seguirá siendo reconocido por la cámara, siempre y cuando no se coloque fuera del campo de visión del sensor.



Figura 5.14. Módulo de captura: reconocimiento de un usuario.

Ya que el usuario haya sido reconocido por el sensor aparecerá una pantalla verde, y en la esquina superior izquierda se mostrará en un recuadro la vista previa del sensor RGBD (Figura 5.15). Dicho recuadro sirve para que el usuario vea si se encuentra posicionado dentro del campo de visión del sensor.



Figura 5.15. Módulo de captura: pantalla de inicio.

En la Figura 5.16 se puede observar cómo es que la marioneta digital responde a los movimientos de la persona que está situada frente al sensor, esto se efectúa en tiempo real.



Figura 5.16. Módulo de captura: movimientos de una persona representados en una marioneta digital.

En la siguiente sección se explica cómo se lleva a cabo el mapeo de los movimientos captados por el sensor en las articulaciones de las marionetas digitales.

#### 5.2.2 Etapa de procesamiento

En la etapa de procesamiento se mapean las articulaciones del esqueleto del sensor con las articulaciones del esqueleto de la marioneta digital. En la Figura 5.7 se mostró cómo está compuesto el esqueleto de 10 puntos para la plataforma FSA-MD, en esta sección se explicará cómo se construyó y definió el esqueleto de la marioneta digital.

#### 5.2.2.1 Estructura general de la marioneta digital

Las marionetas digitales fueron creadas a base de gráficos vectoriales en el formato SVG, debido a que, como se mencionó anteriormente, SVG es un dialecto XML, accesible, declarativo y editable. SVG permite articular y animar una marioneta digital usando Javascript, DOM, CSS o SMIL (*Synchronized Multimedia Integration Language*, Lenguaje de Integración Multimedia Sincronizada), además es importante recordar las características que ofrecen los gráficos vectoriales (véase 3.4.2). Las marionetas digitales están compuestas mayormente de trayectorias o *paths* (Figura 5.17) y agrupaciones (Figura 5.18).



Figura 5.17. Ejemplo de un path en un gráfico SVG.



Figura 5.18. Ejemplo de una agrupación en un gráfico SVG.

En cuanto al esqueleto de 10 puntos, mencionado previamente (Figura 5.7), la marioneta lo incluye y representa como se muestra en la Figura 5.19, esto en correspondencia a las 10 articulaciones de la marioneta. Dichas articulaciones fueron codificadas como nodos específicos en SVG (denominados pivotes) e identificados por su atributo **id** con el valor **pivot\_mx**, donde **x** es el número de pivote, que puede tener un valor entre 0 y 9.



Figura 5.19. Ubicación de pivotes en una marioneta digital.

Como se explicó previamente, las marionetas se componen de *paths*, los cuales representan alguna parte de la marioneta, por ejemplo un brazo o una pierna, sin embargo, no es útil tener

todos los *paths* como elementos independientes. Debido a esto se tienen que agrupar los *paths* que correspondan a la misma sección. En la Tabla 5.2 se muestra cómo deben hacerse las agrupaciones para la mayoría de los casos.

Grupo	Paths
m0	Cara (ojos, nariz, boca, etc.)
m1	Brazo izquierdo (hombro, brazo, etc.)
m2	Antebrazo izquierdo (codo, antebrazo, mano, etc.)
m3	Brazo derecho (hombro, brazo, etc.)
m4	Antebrazo derecho (codo, antebrazo, mano, etc.)
m5	Cadera
m6	Muslo izquierdo
m7	Pierna izquierda (rodilla, tobillo, pie, etc.)
m8	Muslo derecho
m9	Pierna derecha (rodilla, tobillo, pie, etc.)
puppet	Todo

Tabla 5.2.	Estructura	de agrupación	de paths.
------------	------------	---------------	-----------

Cada agrupación se identifica por el valor del atributo id, el cual debe ser **mx**, donde **x** es el número de grupo al que corresponden los *paths* (según la tabla anterior), este número puede tener un valor entre 0 y 9. En la Figura 5.20 se muestra cómo se representaría un grupo en código.



Figura 5.20. Representación en código SVG de un grupo.

En el Apéndice B se explica de una manera más detallada cómo es el proceso de elaboración de una marioneta digital.

#### 5.2.2.2 Mapeo de pivotes en la marioneta digital

Al igual que en el proceso de adquisición (véase 5.2.1), se utilizaron los lenguajes de programación Processing y Java para el mapeo de pivotes en las marionetas digitales, y para manejar el sensor RGBD se usó la librería OpenNI<sup>®</sup> para Kinect<sup>®</sup>.

En la Figura 5.21 se muestra un fragmento del código utilizado para definir el mapeo de pivotes en la marioneta digital. La clase **singleton Puppet** implementa el protocolo **MDP** y representa las articulaciones de la marioneta mediante un arreglo de instancias de la clase **Joint**.

```
class Joint {
    Point pivot; // posición del pivote
    float angle; // ángulo de giro de articulación
    PShape group; // trazos equivalente SVG
    ...
}
class Puppet implements MDP { // Marioneta digital
    ArrayList<Joint> joints; // conjunto de articulaciones
    ...
}
```

Figura 5.21. Representación sintética y simbólica en Java del mapeo de pivotes.

Cada uno de estos objetos **Joint** tiene a su vez un punto de giro (atributo **pivot**), un ángulo de desplazamiento (atributo **angle**) y un grupo de figuras 2D tipo **PShape** que representa el aspecto visual de la marioneta, producto de la importación de un archivo SVG.

#### 5.2.2.3 Generación de movimiento de las articulaciones de la marioneta digital

Para obtener los movimientos deseados se desarrollaron múltiples experimentos. El experimento más básico para controlar una marioneta (clase **Puppet**) previó tres etapas,

primero se obtuvo la información del esqueleto Kinect (OpenNI<sup>®</sup> API), luego se calcularon los desplazamientos angulares de cada articulación y finalmente se mapearon, actualizaron y visualizaron las articulaciones de la marioneta.

Un dato importante a mencionar, es que los movimientos que realiza el usuario solo pueden hacerse en un plano 2D, ya que de lo contrario no son reconocidos por el módulo de captura y no se visualizan. Otro aspecto importante es que el sistema no es capaz de seguir el desplazamiento del usuario, es decir, si el usuario se mueve a otro lugar (dentro del campo de visión del sensor RGBD) el módulo de captura no lo interpretará como un movimiento y por lo tanto la marioneta se mantendrá en su lugar. Para mover la marioneta de lugar se utilizan otras funcionalidades de los reproductores de animación de marionetas (véase 5.3). En Tabla 5.3 se muestran algunos ejemplos de movimientos detectados y no detectados por el sistema.

Movimientos detectados	Movimientos no detectados
Alzar los brazos hacia los lados.	Mover los brazos hacia adelante o atrás.
Mover los pies y piernas a los lados.	Mover las piernas hacia adelante o atrás.
Mover la cabeza a los lados.	Mover la cabeza hacia adelante o atrás.
Balancear el cuerpo a los lados.	Inclinar el cuerpo al frente.
	Caminar en cualquier dirección.
	Saltar.
	Girar el cuerpo.

Tabla 5.3. Ejemplos de movimientos detectados y no detectados por el módulo de captura.

# 5.2.3 Etapa de visualización

En esta etapa se muestra la animación en tiempo real de la marioneta digital. La etapa de visualización es la única etapa que realmente ve el usuario, ya que es donde se muestra cómo es que la marioneta sigue los movimientos que él va realizando. Durante esta etapa solo debe haber una persona situada frente al sensor, no obstante, si el sensor ya identificó a un usuario y otro llega a posicionarse dentro del mismo campo de visión del sensor RGBD, lo único que ocurrirá

es que se dibujará el esqueleto del segundo usuario, pero la marioneta seguirá reproduciendo los movimientos del primer usuario (Figura 5.22).



Figura 5.22. Módulo de captura: caso de visualización con dos usuarios situados frente al sensor RGBD.

Si por algún motivo el usuario principal llega a salir fuera del campo de visión del sensor, la marioneta se mostrará estática en el visualizador, ver Figura 5.23. Además, aunque el usuario secundario permanezca frente al sensor, el módulo de captura actuará como si no hubiera ninguna persona y saldrá del visualizador (Figura 5.24).



Figura 5.23. Módulo de captura: pérdida del reconocimiento del usuario principal.



Figura 5.24. Módulo de captura: pérdida de la imagen por falta del reconocimiento del usuario.

### 5.3 Reproductores de animación de marionetas

Los reproductores de animación de marionetas permiten a los usuarios manipular y visualizar los movimientos generados en el módulo de captura. Un módulo de captura puede tener integrado un reproductor de animación de marionetas, tal es el caso del módulo de captura implementado en Processing.

El reproductor de animación de marionetas se habilita una vez que los datos adquiridos por el módulo de captura llegan a la etapa de visualización. El reproductor cuenta con un panel frontal que se compone de 14 botones, cada uno de los cuales corresponde a una función en particular, las cuales también están asignadas a determinadas teclas del teclado de la computadora. En la Tabla 5.4 se describe el funcionamiento de cada botón.

Botón	Función	Botón	Función
< >	Selección del fondo.	L	Rotación de 90° a la izquierda.
<u>^  </u>	Selección de la marioneta deseada.	R	Rotación de 90° a la derecha.
+	Zoom in.	1	Rotación de 5° a la izquierda.
-	Zoom out.	r	Rotación de 5° a la derecha.
1	Resolución normal.	v	Video de fondo.
2	Resolución 2x.	Х	Mostrar/ocultar vista previa del
			sensor RGBD.

Tabla 5.4. Funciones de los botones del panel frontal.

En la Figura 5.25 se muestran algunas funciones, definidas en el panel frontal, aplicadas a una marioneta digital en el reproductor de animación de marionetas.



Figura 5.25. Funciones del reproductor de animación de marionetas: a) sistema inicial, b) *zoom out*, c) rotación a la derecha.

Para utilizar el reproductor de animación de marionetas existen 2 escenarios, en el primero se necesitan 2 personas, una de ellas para realizar los movimientos y la otra para controlar las funciones de animación con el teclado de la computadora (Figura 5.26).



Figura 5.26. Escenario de dos personas para el uso del reproductor de animación de marionetas.

En el segundo escenario, basta con una persona que realice los movimientos y controle las funciones de animación por medio del panel frontal, esto es posible gracias al uso de un ratón inalámbrico como el Magic Trackpad de Mac (Figura 5.27).



a)



b)

Figura 5.27. Escenario de una persona para el uso del reproductor de animación de marionetas: a) control de funciones de animación, b) realización de movimientos.

## 5.4 Servidor para la fusión de datos de sensores RGBD

Para emplear el esquema del protocolo de red de tiempo real con WebRTC y Socket.IO fue necesario utilizar un servidor web.

La instalación del servidor web se hizo en una computadora Raspberry Pi 2 modelo B (Figura 5.28). Uno de los motivos por los cuales se utilizó esta tarjeta es debido a que WebRTC establece que el intercambio de elementos multimedia debe realizarse a través de un protocolo seguro (dependiendo del tipo de capa esto puede referirse al uso del HTTPS o SSL o TLS), el cual se configura más fácilmente en una tarjeta Raspberry Pi. En la Tabla 5.5 se muestran algunas características de la tarjeta Raspberry Pi 2.



Figura 5.28. Tarjeta Raspberry Pi 2 modelo B.

Tabla 5.5. Características de la tarjeta Raspberry Pi 2, modelo B.

Característica	Descripción
CPU	ARM Cortex-A7 4 núcleos
Velocidad del CPU	900 MHz
Puertos	USB (4), Ethernet (1), Full HDMI (1)
Conectores GPIO	40 pines
<b>Puertos Ethernet</b>	1
Audio	Conector de audio 3.5 mm combinado con video compuesto.
Video	Interface de cámara (CSI), Interface de pantalla (DSI), Núcleo de
	gráficos VideoCore IV 3D
Otros conectores	Tarjeta micro SD

En cuanto al software, se instaló el sistema operativo Raspbian en la versión Jessie con Pixel (Figura 5.29), en el capítulo I del Apéndice C se detallan los pasos a seguir para descargar e instalar el sistema operativo Raspbian en la tarjeta Raspberry Pi.



Figura 5.29. Escritorio del sistema operativo Raspbian con Pixel.

Después de instalar el sistema operativo Raspbian se realizaron algunas configuraciones en la tarjeta de desarrollo para poder trabajar de manera remota, al utilizar una conexión remota se evitó la necesidad de tener periféricos (ratón, teclado y pantalla) conectados a la tarjeta Raspberry Pi. La conexión remota se hizo por medio del protocolo SSH (*Secure Shell*), el cual requirió una IP destino a la cual enlazarla, por este motivo fue importante conocer los tipos de IP que existen. Las IP se pueden clasificar de diferentes formas: privadas o públicas, estáticas o dinámicas, y de acuerdo a su longitud IPv4 o IPv6. En este caso en particular, lo más recomendable fue asignar una IP estática a la tarjeta Raspberry Pi. Ésta se configuró asignando directamente la IP en el enrutador.

Una vez que se asignó una IP estática a la Raspberry Pi, se instaló NodeJS para poder ejecutar el servidor web.

El servidor web se implementó en Javascript, utilizando NodeJS y Express para crear más fácilmente el servidor. Del lado del cliente se trabajó también con Javascript y para el diseño de las páginas web se utilizó el motor de vistas Jade y CSS3. En la Figura 5.30 se muestra el archivo *package.json* utilizado.

```
1
   {
      "name": "fsa md",
      "version": "1.0.0",
 З
 4
      "description": "Protocolo de tiempo real para la animacion de marionetas digitales",
 5
      "main": "server.is".
      "scripts": {
        "test": "echo \"Error: no test specified\" && exit 1",
        "start": "node server.js"
8
9
      }.
10
      "repository": {
        "type": "git",
        "url": "git+https://github.com/egonzalezj/fsa_md.git"
      },
14
      "author": "Elisabet Gonzalez".
      "license": "ISC",
      "bugs": {
17
        "url": "https://github.com/egonzalezj/fsa_md/issues"
18
      }.
19
      "homepage": "https://github.com/egonzalezj/fsa_md#readme",
      "dependencies": {
20
        "express": "^4.14.0",
        "jade": "^1.11.0",
        "socket.io": "^1.7.2"
24
     }
25 }
```

Figura 5.30. Archivo package.json.

Para acceder al servidor a través de un dominio se utilizó el servicio de No-IP, con este servicio se pudo asociar la IP estática de la tarjeta Raspberry Pi con un dominio gratuito. De esta manera, solamente fue necesario abrir un navegador web y escribir la dirección *http://fsa-md.zapto.org* para acceder al servidor web. Luego de haber configurado el *host* del servidor web, éste quedó listo para recibir las solicitudes de los clientes.

Cuando los clientes solicitan un acceso al servidor pueden acceder a la aplicación web de la plataforma FSA-MD (Figura 5.31), la cual cuenta con un diseño responsivo, por lo que se puede adaptar a cualquier dispositivo móvil o PC.



Figura 5.31. Página de inicio de la aplicación web para la plataforma FSA-MD.

Si el usuario navega a través de la página de inicio se encuentra con dos opciones, la primera de ellas permite el acceso a los clientes de captura (Figura 5.32), mientras que la segunda da el acceso a los clientes de visualización (Figura 5.33).



Figura 5.32. Pantalla de acceso a los clientes de captura.



Figura 5.33. Pantalla de acceso a los clientes de visualización.

Los clientes de captura de la aplicación web simulan los módulos de captura implementados en Processing. Para acceder al módulo de captura, el usuario debe presionar el botón Entrar de la Figura 5.32, al hacer esto aparece un formulario de acceso (Figura 5.34).



Figura 5.34. Formulario de acceso para los clientes de captura.
En esta parte de la aplicación se solicitan los datos más indispensables del usuario, los cuales son: el nombre de usuario y la sala a la que se unirá. La razón por la que se piden estos datos es para poder distinguir a los clientes conectados, esto se podría hacer con la IP, sin embargo, puede darse el caso de que una misma IP tenga más de una sesión abierta. Además, la plataforma se diseñó bajo un esquema de chat, donde los clientes entran a salas con otros usuarios y cada uno tiene un nombre de usuario único. Si el usuario no introduce un nombre de usuario, el sistema despliega un mensaje de alerta (Figura 5.35) y no permite el acceso hasta que el usuario haya proporcionado un nombre de usuario.



Figura 5.35. Mensaje de error: no se ha proporcionado un nombre de usuario.

Las salas simulan las sesiones privadas que los usuarios utilizarían para fusionar los datos de captura provenientes de sus sensores RGBD. De hecho, los usuarios pueden crear una sala e invitar colaboradores, no obstante, para esta versión del sistema, se diseñaron solamente 4 salas. Una vez que el usuario provea un nombre de usuario entonces podrá acceder a la sala seleccionada (Figura 5.36). En una sala, los usuarios tienen la posibilidad de ver una lista de los usuarios conectados en esa sala en particular (Figura 5.37).

PLATAFORMA FSA-MD V1.0	
1 usuario	Sala #1
Usuarios	conectados:
Elis	sabet
Datos d	e entrada:
Da	tos
Iniciar captura	Detener captura

Figura 5.36. Sala de producción de animación de marionetas digitales.

PLATAR	FORMA
FSA-M	D V1.0
3 usuarios	Sala #1
Usuarios o	onectados:
Elisa	abet
Da	vid
Sil	via
Datos de	entrada:
Dat	os
Iniciar captura	Detener captura

Figura 5.37. Lista de usuarios conectados.

De igual forma, en la consola donde se ejecuta el servidor se muestra cómo se almacenan los nombres de usuarios de los clientes conectados (Figura 5.38).



Figura 5.38. Consola del servidor web.

### 5.5 Protocolo de red de tiempo real para la fusión de datos multimedia

La transmisión de datos se hizo mediante la implementación de un protocolo de red de tiempo real, el cual utiliza WebRTC y Socket.IO, ésta última es una librería que permite utilizar *websockets* de una manera sencilla.

Para efectos del protocolo, se consideraron los módulos de captura como clientes de captura (CC) y los reproductores de animación de marionetas como clientes de visualización (CV). Debido a las especificaciones de WebRTC y Socket.IO, y siguiendo el modelo de arquitectura TCP/IP, los protocolos que se utilizaron fueron IP para la capa de internet, TCP en la capa de transporte y HTTP en la capa de aplicación.

Como se mencionó en la sección 4.4, WebRTC trabaja en navegadores web, por tal motivo para la implementación del protocolo de red de tiempo real se desarrolló una aplicación web. Para dicho protocolo se utilizó una topología de red tipo estrella, tal y como se aprecia en la Figura 5.39.

Tanto los CC como los CV se construyeron con herramientas de desarrollo web como Jade, CSS3 y Javascript. Para poder realizar el intercambio de mensajes entre el servidor web y los clientes, se incluyó la librería Socket.IO en el lado del servidor y en el lado del cliente (Figura 5.40).



Figura 5.39. Representación del protocolo de red de tiempo real basado en una topología de red tipo estrella.



Figura 5.40. Comunicación cliente-servidor con Socket.IO.

El intercambio de mensajes entre el cliente y el servidor comienza una vez que se hace la apertura de los *websockets*, para esto, el cliente web hace una petición al servidor web para realizar dicha apertura. Por su parte, el servidor web responde a la solicitud de apertura de *websockets* y envía como respuesta un identificador único para el *websocket*. Si otro cliente se conecta al servidor, éste manda una solicitud para abrir un nuevo *websocket*. De igual forma que con el primer cliente web, el servidor responde asignando un identificador único (**socket\_id**), ver Figura 5.41.



Figura 5.41. Diagrama de secuencia de intercambio de mensajes entre dos clientes web con Socket.IO.

Por cada nueva conexión, el servidor envía la variable **numUsers** para dar a conocer, a todos los clientes conectados, el número de usuarios en línea, información que aparece reportada en la lista "Usuarios Conectados" en la pantalla del cliente (Figura 5.37). Una vez conectados los clientes, los *websockets* permanecen abiertos durante toda la sesión de comunicación, esto es de gran utilidad en sistemas de tiempo real porque no es necesario estar mandando solicitudes al

servidor para poder realizar el envío de datos (mejor conocido como *polling*). En lugar de esto, con Socket.IO solo es necesario hacer una solicitud al servidor, después de eso se habilita el envío de datos bajo un esquema de eventos.

La aplicación web desarrollada se asemeja a un sistema de chat, en el cual existen salas que permiten tener un control sobre el tipo y número de usuarios. La idea principal de utilizar salas fue que en cada una de ellas se pudiera habilitar una sesión de animación de marionetas, de manera que cada sesión fuera independiente una de la otra. En la Figura 5.42 se muestra cómo se conectan múltiples clientes, provenientes de diferentes navegadores web, a una sala.



Figura 5.42. Diagrama de secuencia de conexión de múltiples clientes a una sala.

Dentro de la aplicación web, el intercambio de datos se efectúa cuando el usuario ingresa a una sala, introduce los datos de animación e inicia la captura. El área de datos de entrada se ubica debajo de la lista de usuarios, tal como se indica en la Figura 5.43. Para iniciar la captura de datos se debe introducir correctamente la información que se va a transmitir. Cuando se inicia la captura de datos se activa una vista previa de la animación (Figura 5.44), mientras tanto en el servidor se imprimen los datos recibidos (Figura 5.45).



Figura 5.43. Área para datos de entrada.



Figura 5.44. Vista previa de la animación.



Figura 5.45. Recepción de datos de animación en el servidor web.

Con Socket.IO, la comunicación de datos se hace en un modo de operación orientado a conexión, *full-duplex* y multipunto, aunque también es posible punto a punto (*unicast*), ver Figura 5.46.



Figura 5.46. Transmisión punto a punto (unicast) con Socket.IO.

En el modo de transmisión *unicast* se establece un intercambio de mensajes entre cada cliente y el servidor, este tipo de transmisión de datos se utilizó para enviar al servidor web los datos de tipo texto y recibir como respuesta esos mismos datos. Esto permitió que el usuario pudiera tener una vista previa de la animación que estaba efectuando y enviando al servidor.

Socket.IO también ofrece otro escenario de comunicación *unicast*, utilizando la instrucción **socket.to(<socketid>).emit()**, ésta sirve para enviar mensajes a un *socket* en particular. Por medio de este tipo de transmisión de datos se habilitaron las sesiones privadas entre dos usuarios, sin necesidad de crear una sala.

El envío de mensajes en modo multipunto se puede hacer como *broadcast* (Figura 5.47) o *multicast* (Figura 5.48).



Figura 5.47. Transmisión punto a multipunto (broadcast) con Socket.IO.

La transmisión *broadcast* fue muy importante para que todos los usuarios tuvieran la información, sobre el número de conexiones y nombres de usuario, disponible en todo momento. En este caso los mensajes se envían a todos los clientes conectados, excepto al emisor.

Socket.IO maneja otro escenario para la transmisión de datos *broadcast*, para éste se utilizó la instrucción **io.emit()**. La diferencia con respecto a lo mostrado en la Figura 5.47 es que el mensaje se envía a todos los clientes que se encuentren conectados al servidor web, incluyendo al emisor. Esto es útil para el evento de nuevos usuarios conectados, en donde todos los demás miembros de la sesión y el mismo nuevo usuario, deben actualizar la lista de usuarios en línea.



Figura 5.48. Transmisión punto a multipunto (multicast) con Socket.IO.

La transmisión de datos *multicast* se utilizó en las salas de animación. Por medio de *multicast* fue posible enviar mensajes a los usuarios que estuvieran conectados a una sala en particular. Además de esto, Socket.IO contempla otros escenarios. El primero se efectúa con la instrucción **io.in().emit** y permite enviar mensajes a todos los clientes de la sala, incluyendo al emisor. El segundo escenario utiliza la instrucción **socket.to().to().emit** y permite seleccionar varias salas a las cuales se deben mandar los mensajes.

La aplicación web se limitó a trabajar solamente con datos de tipo texto, en donde cada cliente generó un flujo de datos que se envió directamente al servidor. Cabe mencionar que, aunque WebRTC habilita la comunicación entre navegadores, en esta aplicación web, todos los flujos de datos de los CC tuvieron que pasar por el servidor web para poder ser transmitidos a los CV. Esto es sumamente importante, ya que, como se aprecia en la Figura 5.49, el servidor se encargó de realizar las tareas de adquisición, integración, fusión y distribución.



Figura 5.49. Esquema de validación del protocolo de red de tiempo real desarrollado.

Después que los flujos de datos de los CC llegaron al servidor web, éste envió un flujo de datos resultante a los CV (Figura 5.50). La ventaja de que se haya creado una aplicación web es que los CV se pudieron ejecutar en cualquier tipo de dispositivo que tuviera un navegador compatible.



Figura 5.50. Cliente de visualización.

# **VI. RESULTADOS Y CONCLUSIONES**

Para este trabajo de tesis, se llevaron a cabo diversas etapas de desarrollo, tal y como se muestra en la Figura 6.1.



Figura 6.1. Etapas de desarrollo del trabajo de tesis.

Después de desarrollar cada una de estas etapas, se obtuvieron diversos resultados, los cuales fueron de suma importancia para evaluar hasta qué punto se cumplieron los objetivos de este trabajo de tesis. A continuación se describen a detalle los principales resultados obtenidos en cada etapa, incluyendo a su vez, sus respectivas conclusiones y mejoras propuestas.

# 6.1 Producción de marionetas digitales

Esta etapa fue relevante debido a que un diseño o implementación inapropiada o incorrecta puede derivar en serios problemas (incluso impedir la ejecución correcta de las siguientes etapas de desarrollo). Para obtener las condiciones más apropiadas en la elaboración de una marioneta digital, se analizó el método de elaboración de marionetas digitales que existía previamente en el laboratorio de investigación. Dicho método consistía en escribir manualmente código SVG, en un archivo de formato XML para definir la apariencia y estructura de una marioneta digital, esto mediante formas primitivas como círculos, elipses, líneas y curvas de Bézier, entre otras. Después de analizar el método anterior, se encontró que éste no era una opción adecuada para estudiantes con pocos conocimientos en programación. Además, si una persona creaba

manualmente en papel su propio personaje, difícilmente lograría recrearlo en código utilizando solamente formas primitivas representadas en SVG.

En vista de lo anterior, se buscó un software que facilitara la creación de marionetas digitales y que fuera capaz de trabajar con gráficos vectoriales. Para vectorizar un dibujo se propusieron dos métodos, el primero fue redibujar manualmente los trazos tomando como plantilla la imagen digitalizada y el segundo fue utilizar las herramientas de calcado automático que traen algunos programas. En la Tabla 6.1 se resumen los beneficios y limitantes de cada método, los cuales se obtuvieron después de utilizar y analizar cada uno de ellos. Para este caso en particular se utilizó el software Adobe Illustrator<sup>®</sup>.

Método	Beneficios	Limitantes
Manual	✓ El dibujo puede ser dibujado a	★ Es muy tardado, ya que se tiene que
(Redibujo)	color, o en blanco y negro.	rehacer el dibujo, trazo por trazo.
	✓ Es posible corregir imperfecciones	∗ Requiere conocimiento detallado
	del dibujo, o modificar la	sobre manejo de software de
	estructura de éste al momento de	graficación vectorial.
	realizar la vectorización.	
Automático	✓ El proceso de vectorización es	∗ Solo funciona con dibujos en
(Calcado)	significativamente más rápido que	blanco y negro.
	el método manual.	∗ Se propagan y magnifican los
	✓ Requiere poco conocimiento sobre	errores de trazado al momento de
	manejo de software de graficación.	vectorizar el dibujo.

Tabla 6.1 Cuadro com	narativo de los	mátodos utilizados	nara vectorización d	a imánanac
rabia 0.1. Cuadro com	iparativo uc ios	metodos utilizados	para vectorización u	c magenes

Tomando como base los resultados de la tabla anterior, se decidió emplear el método automático. El motivo principal de la decisión fue el tipo de usuarios a los cuales estuvo orientado el programa piloto del presente trabajo, donde participaron tanto estudiantes de licenciatura en administración como de todas las ingenierías. En general, los estudiantes no poseían conocimiento sobre herramientas de software especializadas en la edición de gráficos vectoriales. Una vez que se eligió el método, se realizaron pruebas con distintos programas de edición de gráficos vectoriales para seleccionar el más adecuado. La selección del software se basó principalmente en tres criterios: costo, compatibilidad con versiones de SVG y compatibilidad multiplataforma. En la Tabla 6.2 se muestran los programas de graficación vectorial analizados.

Software	Costo MXN	SO / Plataformas	Compatibilidad
Adobe Illustrator <sup>®</sup>	\$538.00 / mes	Windows®	Soporta SVG 1.1
		Mac OS	
iTrace	\$538.00	Mac OS	No especificada
Super Vectorizer	\$718.15	Mac OS	Soporta SVG 1.0
Inkscape®	\$0.00	Linux	Soporta SVG 1.1
		Windows®	
		Mac OS	

Tabla 6.2. Comparativa entre los distintos paquetes de software para vectorización de imágenes.

Uno de los objetivos de este trabajo de tesis fue desarrollar un sistema de animación de marionetas multiplataforma, esto con la finalidad de permitir, de la mejor forma posible, que los usuarios pudieran crear sus marionetas digitales independientemente de la plataforma computacional usada. Por dicho motivo, se decidió utilizar Inkscape<sup>®</sup> como software multiplataforma de diseño de marionetas digitales, aunado al hecho de ser un programa gratuito y muy eficiente para vectorizar cualquier dibujo a blanco y negro.

Luego de haber establecido el método y herramienta de software para el proceso de vectorización, se encontró que en esta misma herramienta se podían hacer el resto de las etapas de la producción de marionetas digitales. De este modo, se definió el proceso de elaboración de marionetas digitales MD-01, el cual quedó documentado en el Apéndice B. Durante el periodo enero – diciembre 2015, tras realizar 22 nuevas marionetas, se detectó un problema para articular correctamente una marioneta digital. El problema residió principalmente en la manera en la que Inkscape<sup>®</sup> vectorizó los dibujos. Como consecuencia de lo anterior, se desarrolló el procedimiento MD-02, también documentado en [197]. Ambos métodos tienen sus ventajas y desventajas, siendo el tiempo dedicado para crear una marioneta, el factor de mayor relevancia.

Para determinar cuál método fue más rápido y cuáles etapas consumieron más tiempo, se efectuaron pruebas con 19 personas de un rango de edad de los 15 a los 56 años (Figura 6.2). En el Apéndice D, se muestran los datos estadísticos de las muestras tomadas al momento de realizar las pruebas.



Figura 6.2. Voluntarios para las pruebas de validación del procedimiento MD-01.

Aunque existen dos métodos para elaborar una marioneta digital, ambas se componen de las mismas etapas, lo único en que difieren es el orden en el que se lleva a cabo cada una de las actividades. Tomando en cuenta las 19 muestras, se obtuvieron los tiempos promedio que aparecen en la Tabla 6.3. Cabe mencionar que no todas las personas lograron completar todas las etapas, estos casos se omitieron en el cálculo del tiempo promedio por etapa.

Etapa	Tiempo (hrs.)
Trazado de la imagen	00:21:41
Digitalización de la imagen	00:16:58
Vectorización de la imagen	00:20:28
Coloreado de la imagen	00:21:06
Sectorizado de la imagen	00:37:51
Asignación de pivotes	00:33:31
Agrupación	00:35:29
Correcciones	00:14:40

Tabla 6.3. Tiempos promedio para cada etapa del proceso de elaboración de una marioneta digital.

Como se aprecia en los resultados, la etapa que más tiempo consumió fue el sectorizado de la imagen. En el método MD-01 esto se trabaja directamente en el programa Inkscape<sup>®</sup>, donde

por medio de ciertas herramientas del programa es posible separar las partes de la marioneta que así lo requieran. El inconveniente de esto es que para los usuarios era complicado saber en dónde recortar la marioneta, y como se mencionó previamente, fue éste el motivo por el cual surgió el método MD-02. En el método MD-02 no se prescindió de la etapa de sectorizado sino que se realizó físicamente sobre el dibujo en papel en la primera etapa (Figura 6.3).



Figura 6.3. Voluntarios para la pruebas de validación del procedimiento MD-02.

Según lo mostrado en la Figura 6.4, se puede concluir que el método MD-02 consume casi el doble de tiempo que el método MD-01, sin embargo, se deben considerar otros factores como la edad y el grado de estudios porque esto determina la capacidad para manejar el equipo de cómputo y para comprender las instrucciones de naturaleza más técnica y especializada, las cuales están descritas en el manual de usuario del método MD-01.





Con los datos que se tienen registrados hasta el momento es difícil saber con precisión si el método MD-02 es tan tardado como parece, es por ello que se recomienda para casos futuros tomar más muestras con personas de rangos de edades similares para cada método.

En general, más que detectar cuál es el mejor método, lo que se obtuvo al realizar estas pruebas fue la detección oportuna de errores. Al realizar el proceso, algunos usuarios también aportaron mejoras tanto al proceso como a la detección de errores en la documentación. Gracias a esta detección oportuna de errores fue posible mejorar los manuales de usuario. También surgió la propuesta de complementar a futuro la documentación escrita con videos explicativos sobre cada etapa del proceso de elaboración de una marioneta digital, esto en vista de que algunos usuarios experimentaron dificultades para seguir algunas de las instrucciones descritas de forma textual.

En relación con esto, es importante que en posteriores pruebas se le mencione al usuario desde un inicio que existen dos métodos de trabajo (MD-01 y MD-02), cada uno con sus ventajas y desventajas. También sería práctico permitir que el usuario eligiera el método con el cual se sienta más cómodo ya que el contar con una idea clara del proceso puede influir en los tiempos de elaboración de una marioneta digital.

De acuerdo a la Tabla 6.3, las otras dos etapas que más tiempo consumen son la asignación de pivotes y la agrupación. Los principales problemas que tuvieron los usuarios en estas etapas fueron la falta de conocimiento sobre para qué deben realizarse estas etapas y la poca información que había en los manuales sobre los procedimientos y herramientas a utilizar en dichas etapas. Estos problemas ocasionaron que la marioneta funcionara incorrectamente y que se requirieran hacer muchas correcciones.

Como mejoras a futuro se propone detallar de manera muy específica en los manuales cómo deben colocarse los pivotes y cómo se deben crear los grupos de *paths*. También, es de suma importancia aclarar que existe un número mínimo de grupos SVG necesarios para articular de manera exitosa una marioneta digital. Otra de las mejoras es ampliar la especificación de las marionetas digitales para incluir más articulaciones, como manos y facciones faciales. De esta

manera se podría habilitar en un futuro la animación del cuerpo, manos y cara de una marioneta digital, ya que en este trabajo de tesis solo se trabajó con las extremidades principales de la marioneta (véase 5.2.2.1). Para validar tanto la nueva especificación de una marioneta como las guías audiovisuales y las nuevas versiones de los manuales (MD-01 y MD-02), será necesario llevar a cabo nuevas pruebas piloto con estudiantes.

En cuanto a la producción de marionetas digitales se destaca que los principales logros fueron: la definición de dos procedimientos para elaborar una marioneta digital y la documentación que se hizo para esto mismo, teniendo como entregables dos manuales de usuario y una nueva colección compuesta por 22 marionetas digitales.

## 6.2 Producción de narrativas digitales

Una de las maneras de validar la adecuada funcionalidad de las marionetas digitales fue su uso en la producción de narrativas digitales. Para llevar a cabo la producción de narrativas digitales se hicieron pruebas piloto en dos periodos, las primeras fueron en mayo de 2015, y las segundas en noviembre de 2015. En las primeras pruebas piloto se contó con la participación de 9 equipos de estudiantes de la carrera de Licenciatura en Administración del Instituto Tecnológico de Chihuahua (Figura 6.5).



Figura 6.5. Estudiantes durante las pruebas piloto de mayo de 2015.

A cada uno de los equipos se le asignó una sesión de una hora para cada una de las etapas de producción (grabación de animación, grabación de diálogos y edición), sin embargo este tiempo no fue suficiente ya que en promedio todos los equipos requirieron de al menos 5 horas para producir su narrativa digital. La etapa en la que más invirtieron tiempo fue en la grabación de

las animaciones, la cual se hizo mediante el módulo de captura implementado en Processing (véase 5.2). La razón por la que requirieron de tanto tiempo fue que los estudiantes no estaban familiarizados con tecnologías como los sensores RGBD, hecho por el cual, como medida preventiva, se había solicitado a los estudiantes que acudieran al laboratorio a conocer el funcionamiento del módulo de captura, pero no todos lo hicieron. De esta manera, al iniciar las sesiones de trabajo para la producción de las narrativas digitales, se pudo apreciar la diferencia en tiempos de realización de una narrativa digital en aquellos equipos que sí habían probado previamente el software de captura con sensores RGBD y los que no lo habían hecho.

Otro factor a considerar es que no todas las animaciones que los estudiantes habían propuesto en su guion gráfico se podían realizar, esto debido a las limitantes de duración establecida para la narrativa digital. Por este motivo, la mayoría de los equipos volvió a retrabajar su guion gráfico, lo que les llevó a invertir más tiempo en la grabación de animaciones. Cabe mencionar que los equipos que probaron el software con anticipación, pudieron retrabajar más fácilmente la coreografía de sus escenas. Para las próximas pruebas piloto será importante designar una sesión de trabajo únicamente para conocer el software de captura y modificar las coreografías de las escenas que así lo requieran.

En las primeras pruebas piloto se utilizó una versión del módulo de captura que permite a los usuarios colocar la imagen de fondo que necesiten. Después de que los 9 equipos trabajaran con esta versión se encontró que los estudiantes cometían errores en la grabación de las animaciones, debido a que en ocasiones colocaban una imagen de fondo equivocada que no correspondía a la escena. Estos errores no tuvieron solución más que volver a grabar la animación. Para evitar este tipo de errores se modificó el módulo de captura, de manera que los usuarios no tuvieran que colocar las imágenes de fondo, sino que trabajaran bajo el esquema de pantalla cromática (verde o azul).

En el segundo periodo de pruebas piloto se tuvo la participación de 6 equipos, dando un total de 23 alumnos de las carreras de Ingeniería Geológica e Ingeniería Aeroespacial de la Universidad Autónoma de Chihuahua (Figura 6.6). Estos equipos utilizaron el módulo de captura con el esquema de pantalla verde, lo cual ayudó a reducir los tiempos de grabación de

animaciones. Sin embargo se incrementó el trabajo en la etapa de edición, ya que la inclusión de las imágenes de fondo se hizo en el programa de edición de video. En cuanto a los programas de edición de video, se descubrió la limitante de que no todos los programas permiten trabajar con grabaciones en pantalla verde. Otra limitante fue que se tuvieron que cambiar los colores de algunas marionetas, ya que si alguna de ellas contenía el color verde o azul generaba problemas en la edición con pantalla cromática, por lo tanto no debe haber marionetas con estos colores.



Figura 6.6. Estudiantes durante las pruebas piloto de noviembre de 2015.

Durante el trabajo de producción de narrativas digitales, algunos equipos requirieron utilizar objetos dentro de la animación, esto para simular acciones con la marioneta digital como tomar o lanzar objetos. Debido a que el módulo de captura no considera este tipo de casos de animación, se tuvieron que rediseñar las marionetas digitales para agregar el objeto en ella (Figura 6.7), esta solución no es muy práctica ya que en ocasiones es necesario que el objeto esté en una posición diferente a la de la marioneta. Como una posible mejora se propone asignar un pivote en el centro del objeto o inclusive trabajar con realidad aumentada para poder interactuar con objetos físicos que puedan ser representados por gráficos SVG.

En conclusión, es mejor utilizar la pantalla cromática para grabar las animaciones, e insertar luego las imágenes de las escenas en la etapa de edición. Sin embargo se debe tener presente que el software de edición de video debe ser capaz de trabajar con esta técnica. También se concluye que aunque la producción de narrativas digitales con marionetas digitales es más rápida que con otros métodos, se necesitan varias sesiones (aproximadamente 5 horas) para lograr desarrollar una narrativa digital.



Figura 6.7. Marioneta rediseñada: a) marioneta original, b) marioneta con objeto integrado.

### 6.3 Diseño e implementación de los módulos de captura

Como se explicó en capítulos anteriores, los módulos de captura se encargan de recibir los datos de los sensores RGBD que posteriormente se procesan para ser enviados al servidor. En un inicio se contaba con un módulo de captura implementado en Processing, el cual se fue modificando para adaptarlo a los propósitos de este trabajo de tesis. Por este motivo se desarrollaron distintas versiones, cada una con sus respectivas mejoras y limitantes. En la Tabla 6.4 se muestra el historial de versiones con los cambios más relevantes de cada una de ellas.

En la sección anterior, se mencionó que se hicieron dos periodos de pruebas piloto con diferentes versiones del módulo de captura implementado en Processing, dichas versiones corresponden a la 12a (primer periodo) y 13a (segundo periodo). Aunque el módulo de captura de Processing funcionó adecuadamente en las pruebas piloto, no fue la mejor opción para utilizarlo en la implementación del protocolo de tiempo real, debido a que para integrar WebRTC en una aplicación nativa se tenían que utilizar algunas APIs y librerías, lo cual implicaba un largo proceso de desarrollo.

Para la implementación del protocolo de tiempo real, se adaptó el módulo de captura de Processing, a una aplicación web con un número limitado de funciones. Se encontró que fue más fácil y rápido adaptar el módulo de captura al lenguaje de Javascript aunque esto implicara portar el código, debido a que tanto WebRTC como el servidor web utilizaron este mismo lenguaje para su implementación.

Versión	Mejoras
11	Versión inicial.
12a	Se añade la tecla X/x para ocultar la vista previa del sensor RGBD.
12b	Se añade la tecla B/b para la función de pantalla verde/azul.
12c	Se integran las versiones 12a y 12b.
	Se elimina el uso de la tecla B/b y se agrega el fondo verde/azul a la lista de
	imágenes de fondo.
13	Se agrega un panel frontal para ejecutar los comandos de animación directamente
	en la interfaz.
13a	Se integran las versiones 12c y 13.
	Se elimina el uso de la tecla C/c para ocultar el cursor y se muestra solamente en
	el área del panel.
14	Se agrega la tecla 3 para la función de pantalla completa.

Tabla 6.4. Historial de versiones del módulo de captura implementado en Processing.

Hasta el momento, el módulo de captura implementado en Javascript no cuenta con todas las funcionalidades que ofrece el módulo de captura de Processing. En la actualidad, la herramienta más práctica y sencilla que existe para migrar el código del módulo de captura a Javascript es P5js, una librería de Javascript del lado del cliente basada en los principios de Processing. Uno de los trabajos a futuro es terminar la adaptación del módulo de captura a Javascript utilizando P5js. La limitante de este módulo de captura es que solo funciona con datos de tipo texto que simulan los datos provenientes de los sensores RGBD.

En cuanto al módulo de captura de Processing, se busca que en próximas versiones sea posible ampliar el número de usuarios activos para que puedan interactuar dos o más personas de manera simultánea en el mismo campo de visión del sensor. Para esto será importante separar los flujos de datos de cada usuario de manera que actúen como dos sensores independientes, de este modo el servidor web podría reconocerlos como clientes distintos.

Otro de los trabajos a futuro es completar la integración de la API de la cara e incluir comandos de voz para controlar el comportamiento de la animación en el módulo de captura.

Un aspecto importante a considerar en cualquier tipo de software que se desarrolla es la experiencia de usuario, que determina cómo se siente el usuario al interactuar con las aplicaciones de software. Para evaluar este factor, se propone que se apliquen pruebas UXA (*User Experience Assessment*) en las próximas pruebas piloto.

### 6.4 Implementación del servidor web

Actualmente existen diversas herramientas que facilitan las pruebas de rendimiento, carga y estrés. Una de las herramientas que se utilizaron para obtener las métricas de rendimiento fue Monitis [198]. Dentro de las métricas que fue posible generar se encuentran las siguientes: RPS (*Requests per Second*), *Error Rates*, ART (*Average Response Times*), PRT (*Peak Response Times*), Uptime, uso de CPU y uso de memoria.

Para poder crear el servidor web, se encontraron algunas limitantes, una de ellas fue la dificultad para obtener una IP pública estática. Anteriormente cualquier persona podía crear su propio servidor web y acceder a él desde otra red, sin embargo, en la actualidad los proveedores de internet han configurado sus redes de un modo en que dificultan la creación de servidores web. La solución a este inconveniente es solicitar un servicio de IP homologada, el cual tiene un costo de \$600 MXN por mes. Para el caso de este trabajo de tesis, no fue viable solicitar el servicio de IP homologada ya que se podía trabajar de igual forma dentro de la red local, sin afectar el funcionamiento del sistema. Fue por esta razón que las pruebas de este trabajo de tesis se realizaron con clientes que estuvieron conectados a la misma red.

Para no depender de una computadora de escritorio se montó el servidor web en una tarjeta de desarrollo Raspberry Pi, cabe mencionar que el proceso de instalación no fue tan sencillo como se había considerado en un inicio debido a todas las configuraciones que se deben realizar.

Además, se debe tener en cuenta que no es muy recomendable utilizar este tipo de tarjetas para servidores que requieran un gran almacenamiento de recursos, ya que la memoria de almacenamiento es de baja capacidad, una posible solución sería emplear un disco duro externo de gran capacidad. Para el caso de este trabajo de tesis no se tuvieron problemas con el almacenamiento, debido a que no se guardó una gran cantidad de datos, sino más bien el servidor se centró en procesar información. Dicho esto, es importante mencionar también que otros de los factores que podrían haber limitado el rendimiento del sistema son la capacidad de memoria RAM que soporta la tarjeta Raspberry Pi y la velocidad de su procesador. Esto implica que no es recomendable utilizar este tipo de tarjetas para servidores que requieran un alto consumo de procesos simultáneos.

### 6.5 Implementación del protocolo de tiempo real

Al trabajar con WebRTC para la implementación del protocolo de tiempo real, se tuvieron que utilizar algunos elementos de HTML5 para transmisión multimedia como *getUserMedia*. Luego de implementar dichos elementos y ejecutarlos en el servidor se encontró con la limitante de que no es posible utilizarlos en protocolos no seguros. Este hecho fue un gran inconveniente ya que, inicialmente, el servidor web se instaló en la red del Instituto Tecnológico de Chihuahua y esta red no cuenta con un protocolo seguro (HTTPS). Además, para que un protocolo HTTP sea seguro se debe instalar un certificado de seguridad SSL, el cual tiene un costo. Como solución a lo anterior se instaló el servidor localmente en una computadora del laboratorio de M-Learning del I. T. de Chihuahua para requerir el uso de la red de la institución, sin embargo ciertas funciones de WebRTC no se pudieron ejecutar de manera local por lo que se buscaron otras alternativas.

Como se mencionó en la sección anterior, el servidor finalmente se instaló en una red residencial en donde se pudiera tener un mejor control de las configuraciones de red de manera que se pudiera instalar un certificado SSL. En cuanto al certificado de seguridad, los costos que manejan algunas compañías como GoDaddy son de \$83.75 MXN. Aunque los precios no son muy elevados, se encontraron otros sitios que ofrecen certificados gratuitos, sin embargo, lo que ofrecen estas compañías es solamente un periodo de prueba de 30 días. En vista de las

complicaciones que creó el hecho de no contar con un protocolo seguro se buscó la manera de implementar el protocolo sin necesidad de utilizar funciones que requirieran del protocolo seguro. Como solución se decidió trabajar únicamente con el envío de texto y así evitar el uso de la función *getUserMedia*, la cual sirve para soportar audio y video en el navegador web.

Previamente se señaló que WebRTC es soportada por varios navegadores, no obstante, fue una limitante que navegadores como Safari, o algunas versiones de los navegadores Chrome y Opera, no soportaran WebRTC, ya que uno de los principales objetivos de esta tesis era contar con un sistema multiplataforma.

Aunque WebRTC es un *framework* muy útil para el protocolo que se implementó, fue necesario incluir otras librerías como Socket.IO. El trabajar con ambas herramientas facilitó el intercambio de mensajes del protocolo propuesto, ya que por una parte, WebRTC se dedicó a la administración de las sesiones, mientras que Socket.IO se encargó de manejar los eventos de envío de datos, todo esto en tiempo real.

Después de haber empleado WebRTC en el protocolo de tiempo real, se pudo concluir que es una herramienta útil y novedosa para la transmisión multimedia en tiempo real, no obstante también se comprobó el hecho que en un protocolo no se puede utilizar solamente WebRTC sino que requiere de un mecanismo para manejar el intercambio de mensajes, tal como lo menciona la documentación oficial de WebRTC.

Los resultados de la implementación del protocolo solo se hicieron con simulaciones de clientes de captura, enviando cadenas de texto al servidor, representando los vectores de posiciones que se obtienen de un sensor RGBD.

Como trabajo futuro se propone utilizar una base de datos, que esté optimizada para trabajar con protocolos de tiempo real, para almacenar una lista de los usuarios que se encuentren en línea. En esta base de datos también se guardarían los datos del cliente como el tipo de red, proveedor de internet, región e IP, entre otros. Otra de las mejoras que se proponen es crear un repositorio de marionetas en donde cada una de ellas tenga correctamente asignados sus respectivos metadatos, esto ayudaría a poder crear un sistema de búsqueda de marionetas, ya sea por nombre, autor, tipo, etc.

En la versión actual la transmisión no realiza ningún tipo de almacenamiento de datos, es por ello que otra de las mejoras que se pueden realizar es agregar la función de almacenar las animaciones de marionetas para que el usuario pueda tener disponible funciones como reproducir el material *offline* o retroceder el tiempo de una transmisión en vivo, en el caso de que dicho usuario se uniera a la transmisión una vez que ésta ya haya comenzado.

Dado que la validación del protocolo solo se hizo con simulaciones de los clientes de captura, el siguiente paso debe consistir en transmitir los datos reales de los sensores RGBD. Para esto será necesario diseñar un adaptador para los sensores RGBD. Con estos adaptadores se busca leer los datos de cada cámara, como la resolución y el número de FPS al cual funciona el sensor, para que los clientes de captura no se vean afectados cada vez que se utilice una cámara distinta, sino que puedan leer la información de los sensores por medio del adaptador.

### 6.6 Animación de marionetas digitales mediante la plataforma FSA-MD

Como ya se explicó anteriormente, la animación de marionetas digitales en tiempo real solamente se hizo con datos simulados, representando tanto a los clientes de captura como a las marionetas digitales. Para validar el sistema se implementó la API de la marioneta [200] en la aplicación web desarrollada para ejecutar el servidor web, esto se hizo utilizando ProcessingJS. La estructura de los datos de entrada para la animación de la marioneta estuvo compuesta por un arreglo de dos dimensiones, en el cual los arreglos tuvieron un tamaño de 3 y un tipo de dato **int**, esto es, **[int id, int steps, int ang]**, donde **id** fue el número de pivote, **steps** el número de pasos que se movería la articulación y **ang** el ángulo de desplazamiento.

Para el envío de datos se estableció primeramente un intervalo de tiempo de captura de datos de 100 ms., esto quiere decir que cada 100 ms. se estuvieron leyendo y almacenando los datos provenientes de los clientes. Como medida preventiva, antes de enviar los datos al servidor, se almacenaron en un *buffer*, de manera que cada 100 ms. se llenaba el *buffer* y se iban eliminando

los elementos consumidos para hacer la animación. En cuanto a la fusión de datos, se hizo una prueba en donde se utilizaron dos clientes, uno de ellos para animar las extremidades superiores de la marioneta y el otro para las extremidades inferiores. Los resultados demostraron que se pudieron recibir de manera efectiva los datos provenientes de cada cliente, logrando animar tanto las extremidades superiores de la marioneta digital como las inferiores, todo esto de manera simultánea.

La siguiente validación debe realizarse conectando al menos dos sensores RGBD a un módulo de captura para cada sensor, para esto, primeramente se debe integrar el protocolo de tiempo real al módulo de captura de Processing. Para el envío de datos, es indispensable definir un arreglo con las posiciones de los movimientos generados por el sensor RGBD, y convertirlo a un tipo de dato de texto.

# REFERENCIAS

- [1] W. L. Leite, M. Svinicki and Y. Shi, "Attemped Validation of the Scores of the VARK: Learning Styles Inventory with Multitrait-Multimethod Confirmatory Factor Analysis Models," *Educational and Psychological Measurement*, vol. 70, no. 2, pp. 323-339, Apr. 2010.
- [2] J. A. Ocaña, *Mapas Mentales y Estilos de Aprendizaje*. San Vicente, España: Editorial Club Universitario, 2010.
- [3] E. Martínez Marrero, "Estilos de Aprendizaje," en *Cómo Estudiar Anatomía*, Barranquilla, Colombia: Editorial Universidad del Norte, 2012, ch. 3.
- [4] J. F. Weng, H. L. Kuo and S. S. Tseng, "Interactive Storytelling for Elementary School Nature Science Education," *Advanced Learning Technologies (ICALT), 2011 11th IEEE International Conference* on, pp. 336-338, Jul. 2011.
- [5] A. Pacheco González, M. Ramírez Rojas y C. L. Guzmán González, "Marionetas Digitales: Tecnología Emergente para Narrar Historias con Personajes Animados Mediante la Captura Digital de Movimiento," en *Tecnologías Emergentes en la Educación*, 1 ed. Distrito Federal, México: Pearson, 2015, cap. 11, pp. 161-175.
- [6] A. Pacheco González, I. Robledo Vega and E. González Juárez, "Plataforma de Fusión de Sensores RGB-D para la Animación de Marionetas Digitales," *Revista de Tecnologías Emergentes en la Educación / REMIAM*, vol. 1, no. 1, pp. 101-106, Jun. 2016.
- [7] W. Tsou, W. Wang and Y. Tzeng, "Applying a Multimedia Storytelling Website in Foreign Language Learning," *Computers & Education*, vol. 47, no. 1, pp. 17-28, Aug. 2006.
- [8] O. Balet, "INSCAPE: An Authoring Platform for Interactive Storytelling," in Virtual Storytelling. Using Virtual Reality Technologies for Storytelling, Berlín, Germany: Springer Berlin Heidelberg, 2007, pp. 176-177.
- [9] S. Göbel, L. Salvatore and R. Konrad, "StoryTec: A Digital Storytelling Platform for the Authoring and Experiencing of Interactive and Non-Linear Stories," *Automated*

solutions for Cross Media Content and Multi-channel Distribution, 2008. AXMEDIS '08. International Conference on, pp. 103-110, Nov. 2008.

- [10] B. M. Landry and M. Guzdial, "iTell: Supporting Retrospective Storytelling with Digital Photos," *Proceedings of the 6th ACM conference on Designing Interactive systems - DIS '06*, pp. 160-168, Jun. 2006.
- [11] F. Lu *et al.*, "ShadowStory: Creative and Collaborative Digital Storytelling Inspired by Cultural Heritage," *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11*, pp. 1919-1928, May 2011.
- [12] P. Alavesa, T. Ojala and D. Zanni, "Props: 3D-Game-Like Mediator for Improvisational Storytelling," *Entertaiment Computing*, vol. 5, no. 4, pp. 381-390, Oct. 2014.
- [13] S. H. Minelli *et al.*, "MOVIO: A Toolkit for Creating Curated Digital Exhibitions," *Procedia Computer Science*, vol. 38, pp. 28-33, 2014.
- [14] I. Spitalnik and D. Zaturansky. *PowToon*. (2012). [Online].Available: https://www.powtoon.com, Accessed on: Dec. 5, 2016.
- [15] J. Multisilta and M. Suominen, "MoViE: Mobile Video Experience," Proceedings of the 13th International MindTrek Conference: Everyday Life in the Ubiquitous Era on - MindTrek '09, pp. 157-161, Oct. 2009.
- [16] D. M. Frohlich *et al.*, "StoryBank: Mobile Digital Storytelling in a Development Context," *Proceedings of the 27th international conference on Human factors in computing systems - CHI 09*, pp. 1761-1770, Apr. 2009.
- [17] Storie Inc. *Storie*. [Online]. Available: https://itunes.apple.com/us/app/storie-videostorytelling/id950300668?mt=8, Accessed on: Nov. 24, 2016.
- [18] Zuidsoft. Story Dice. [Online]. Available: https://itunes.apple.com/us/app/story-dicestory-telling/id1105668334?mt=8, Accessed on: Nov. 24, 2016.
- [19] Zuidsoft. Story Dice. [Online].
  Available: https://play.google.com/store/apps/details?id=com.zuidsoft.storystones, Accessed on: Nov. 24, 2016.

- [20] Mombo Labs. *Steller*. [Online]. Available: https://steller.co/, Accessed on: Nov. 24, 2016.
- [21] Tellagami<sup>®</sup> Labs Inc. *Tellagami*. [Online]. Available: https://tellagami.com/, Accessed on: Nov. 24, 2016.
- [22] AFAR Interactive, Inc. Pixotale. [Online]. Available: https://pixotale.com/, Accessed on: Nov. 24, 2016.
- [23] Phases. *Bevola*. [Online]. Available: http://bevola-storytelling.appstor.io/es/, Accessed on: Nov. 24, 2016.
- [24] MyAlbum. [Online]. Available on: https://myalbum.com/es, Accessed on: Nov. 24, 2016.
- [25] Scribz. [Online]. Available: https://itunes.apple.com/mx/app/scribz-art-ofstorytelling/id583834100?mt=8, Accessed on: Nov. 24, 2016.
- [26] Andreas Garzotto GmbH. *Blauzahn*. [Online].Available: http://www.blauzahn.ch./blauzahn/, Accessed on: Nov. 24, 2016.
- [27] Streamlife Concepts Inc. *Dayo*. [Online]. Available: https://www.dayoapp.co/, Accessed on: Nov. 24, 2016.
- [28] Indigital Pty Ltd. Indigital. [Online]. Available: http://www.indigital.net.au/, Accessed on: Nov. 24, 2016.
- [29] Shadow Puppet Inc. Shadow Puppet. [Online]. Available: http://get-puppet.co/, Accessed on: Nov. 24, 2016.
- [30] Allwomenstalk. *Fresh*. [Online]. Available: https://itunes.apple.com/us/app/fresh-storytelling-on-the-go/id913347735?mt=8, Accessed on: Nov. 24, 2016.
- [31] A. Pacheco. *Episodium*. [Online].
  Available: https://itunes.apple.com/us/app/episodium/id678062784?l=es&mt=8,
  Accessed on: Nov. 22, 2016.
- [32] Adobe Systems Software Ireland Ltd. *Adobe Photoshop CC*. [Online]. Available: http://www.adobe.com/mx/products/photoshop.html, Accessed on: Nov. 21, 2016.
- [33] Adobe Systems Software Ireland Ltd. *Adobe Illustrator CC*. [Online]. Available: http://www.adobe.com/mx/products/illustrator.html, Accessed on: Nov. 21, 2016.

- [34] Corel Corporation. *CorelDRAW*. [Online].
  Available: http://www.coreldraw.com/la/product/diseno-grafico-creativo/, Accessed on: Nov. 21, 2016.
- [35] Autodesk, Inc. Autodesk Graphic. [Online]. Available: https://graphic.com/, Accessed on: Nov. 21, 2016.
- [36] The GIMP Team. GIMP. [Online]. Available: https://www.gimp.org/, Accessed on: Nov. 21, 2016.
- [37] Inkscape. Inkscape. (1999). [Online]. Available: https://inkscape.org, Accessed on: Nov. 21, 2016.
- [38] Skencil. Skencil. (1998). [Online]. Available: http://www.skencil.org/, Accessed on: Nov. 21, 2016.
- [39] Xara Group Ltd. Xara LX. (2005). [Online]. Available: http://www.xaralx.org/, Accessed on: Nov. 21, 2016.
- [40] IndiaMART InterMESH Ltd. Indiamart. [Online]. Available: http://www.indiamart.com/vencer-creation/character-designing-services.html#2dcharacter-designing, Accessed on: Dec. 1, 2016.
- [41] L. Leite and V. Orvalho, "Shape Your Body: Control a Virtual Silhouette Using Body Motion," Proceedings of the 2012 ACM Annual Conference Extended Abstracts on Human Factors in Computing Systems Extended Abstracts - CHI EA '12, pp. 1913-1918, May 2012.
- [42] L. Leite and V. Orvalho, "Inter-Acting: Understanding Interaction with Performance-Driven Puppets Using Low-Cost Optical Motion Capture Device," *International Journal of Advanced Computer Science*, vol. 3, no. 2, pp. 65-69, Feb. 2013.
- [43] M. Antonijoan, D. Soler and D. Miralles, "Avatoys: Sistema Híbrido de Marionetas Reales y Digitales," 2014 9th Iberian Conference on Information Systems and Technologies (CISTI), pp. 1-6, Jun. 2014.
- [44] J. R. Parker and C. Martini, "Puppetry of the Pixel: Producing Live Theatre in Virtual Spaces," 2011 IEEE Consumer Communications and Networking Conference (CCNC), pp. 327-331, Jan. 2011.

- [45] J. Gutiérrez Lira, "Técnicas de Animación en 3D y Efectos Especiales," Tesis de Ingeniería, Depto. Ciencias Básicas e Ingeniería. Universidad Autónoma del Estado de Hidalgo, 2006.
- [46] P. Domínguez. *La Animación Definición y Tipos*. (2016). [Online]. Available: https://www.youtube.com/watch?v=H2jqm457sZI, Accesed on: Nov. 25, 2016.
- [47] K. A. Priebe, *The Art of Stop-Motion Animation*. Boston, MA, USA: Thomson Course Technology PTR, 2007.
- [48] C. Barnes *et al.*, "Video Puppetry: A Performative Interface for Cutout Animation," *ACM Transactions on Graphics*, vol. 27, no. 5, pp. 1-9, Dec. 2008.
- [49] M. Dima, J. Lee and M. Wright, "Exploration of Direct Bi-Manual Interaction in Digitally Mediated Stop-Motion Animation," *Proceedings of the 6th Nordic Conference on Human-Computer Interaction Extending Boundaries - NordiCHI '10*, pp. 635-638, Oct. 2010.
- [50] F. Zilly, M. Ziegler, J. Keinert, M. Schöberl and S. Foessel, "Computational Imaging for Stop-Motion Animated Video Productions," *SMPTE Motion Imaging Journal*, vol. 125, no. 1, pp. 42-47, Feb. 2016.
- [51] P. Zolliker, Z. Baranczuk, D. Küpper, I. Sprow and T. Stamm, "Creating HDR Video Content for Visual Quality Assessment," *Signal Processing Conference (EUSIPCO)*, 2013 Proceedings of the 21st European, pp. 1-5, Sep. 2013.
- [52] X. Han, H. Fu, H. Zheng, L. Liu and J. Wang, "A Video-Based System for Hand-Driven Stop-Motion Animation," *IEEE Computer Graphics and Applications*, vol. 33, no. 6, pp. 70-81, Dec. 2013.
- [53] S. Piman and A. Z. Talib, "Puppet Modeling for Real-Time and Interactive Virtual Shadow Puppet Play," 2012 Second International Conference on Digital Information and Communication Technology and it's Applications (DICTAP), pp. 110-114, May 2012.
- [54] T. Igarashi, T. Moscovich and J. F. Hughes, "Spatial Keyframing for Performance-Driven Animation," *Proceedings of the 2005 ACM SIGGRAPH/Eurographics* symposium on Computer animation - SCA '05, pp. 107-115, Jul. 2005.

- [55] Y. Zhuang, Y. Pan and J. Xiao, A Modern Approach to Intelligent Animation. Berlin, Germany: Springer Berlin Heidelberg, 2008.
- [56] Smith Micro Software Inc. *Moho<sup>®</sup> Pro 12*. [Online].
  Available: http://my.smithmicro.com/anime-studio-pro.html, Accessed on: Nov. 25, 2016.
- [57] Toon Boom Animation Inc. *Toon Boom Harmony*. [Online]. Available: http://www.toonboom.com/products/harmony, Accessed on: Nov. 25, 2016.
- [58] Synfig Studio Development Team. *Synfig Studio*. [Online]. Available: http://www.synfig.org/cms/, Accessed on: Nov. 25, 2016.
- [59] Adobe Systems Software Ireland Ltd. *Adobe After Effects CC*. [Online]. Available: http://www.adobe.com/mx/products/aftereffects.html, Accessed on: Nov. 25, 2016.
- [60] Adobe Systems Software Ireland Ltd. *Adobe Animate CC*. [Online]. Available: http://www.adobe.com/mx/products/animate.html, Accessed on: Nov. 25, 2016.
- [61] CELSYS, Inc. *Clip Studio Paint* [Online].
  Available: http://www.clipstudio.net/es/promotion/animation, Accessed on: Nov. 25, 2016.
- [62] A. Mazalek and M. Nitsche, "Tangible Interfaces for Real-Time 3D Virtual Environments," *Proceedings of the international conference on Advances in computer entertainment technology - ACE '07*, vol. 203, pp. 155-162, Jun. 2007.
- [63] N. Hiroki, N. Pantuwong and M. Sugimoto, "A Puppet Interface for the Development of an Intuitive Computer Animation System," *International Conference on Pattern Recognition*, pp. 3136-3139, Nov. 2012.
- [64] J. L. Lin, H. C. Chuan, P. C. Kuan and S. M. Ni, "Assessment of Range of Joint Motion Using Kinect," in *Bridging Research and Good Practices towards Patients Welfare: Proceedings of the 4th International Conference on Healthcare Ergonomics and Patient Safety (HEPS), Taipei, Taiwan, 23-26 June 2014*, London, England: Taylor & Francis Gruop, 2015, pp. 387-392.
- [65] S. Das *et al.*, "Quantitative Measurement of Motor Symptoms in Parkinson's Disease:A Study with Full-Body Motion Capture Data," 2011 Annual International

*Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 6789-6792, Aug. 2011.

- [66] A. Ozgur, S. Bonardi, M. Vespignani, R. Mockel and A. J. Ijspeert, "Natural User Interface for Roombots," *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, pp. 12-17, Aug. 2014.
- [67] S. W. Choi, W. J. Kim and C. H. Lee, "Interactive Display Robot: Projector Robot with Natural User Interface," 2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pp. 109-110, Mar. 2013.
- [68] J. A. Peña Palacio and S. Londoño Arevalo, "Control de Vuelo a Distancia de un Helicóptero RC Mediante una Interfaz de Usuario Natural," 2014 9th Iberian Conference on Information Systems and Technologies (CISTI), Jun. 2014.
- [69] W. Bholsithi, N. Wongwaen and C. Sinthanayothin, "3D Avatar Developments in Real Time and Accuracy Assessments," 2014 International Computer Science and Engineering Conference (ICSEC), pp. 85-90, August 2014.
- [70] M. Z. Patoli, M. Gkion, P. Newbury and M. White, "Real Time Online Motion Capture for Entertainment Applications," *Digital Game and Intelligent Toy Enhanced Learning (DIGITEL), 2010 Third IEEE International Conference on*, pp. 139-145, Apr. 2010.
- [71] Unity Technologies. Unity. [Online]. Available: https://unity3d.com/es/, Accessed on: Nov. 25, 2016.
- [72] J. Pan and J.J. Zhang, "Sketch-Based Skeleton Driven 2D Animation and Motion Capture," in *Transactions on Edutainment VI, LNCS*, Berlin, Germany: Springer, Berlin, Heidelberg, 2009, pp. 164-181.
- [73] A. Pacheco and R. Cruz, "Animating 2D Digital Puppets Using Motion Capture and Vectorial Graphics," *The Graphical Web: Visual Storytelling*, 2014.
- [74] Blender Foundation. *Blender*. [Online]. Available: https://www.blender.org, Accessed on: Nov. 25, 2016.
- [75] Autodesk Inc. Maya. [Online].

Available: http://www.autodesk.mx/products/maya/overview, Accessed on: Nov. 25, 2016.

- [76] Autodesk Inc. *MotionBuilder*. [Online].
  Available: http://www.autodesk.com/products/motionbuilder/overview, Accessed on: Nov. 25, 2016.
- [77] Y. Seol, C. O'Sullivan and J. Lee, "Creature Features: Online Motion Puppetry for Non-Human Characters," *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation - SCA '13*, pp. 213-221, Jul. 2013.
- [78] A. Fender, J. Müller and D. Lindlbauer, "Creature Teacher: A Performance-Based Animation System for Creating Cyclic Movements," *Proceedings of the 3rd ACM Symposium on Spatial User Interaction - SUI '15*, pp. 113-122, Aug. 2015.
- [79] T. Shiratori, M. Mahler, W. Trezevant and J. K. Hodgins, "Expressing Animated Performances through Puppeteering," 2013 IEEE Symposium on 3D User Interfaces (3DUI), pp. 59-66, Mar. 2013.
- [80] E. M. Mokhtar, "Puppetry-Inspired Animation: A Midair Hand Gestures Manipulation for 3D Character Animation," *Proceedings of HCI 2013: The 27th International British Computer Society Human Computer Interaction Conference*, pp. 1-4, Sep. 2013.
- [81] C. Curtis, D. Eisenmann, R. E. Guerrab and S. Stafford, "The Making of Pearl, a 360°
  Google Spotlight Story," ACM SIGGRAPH 2016 VR Village on SIGGRAPH '16, p. 1, Jul. 2016.
- [82] N. Wang, X. Chen, G. Song and H. Parsaei, "A Novel Real-Time Video Transmission Approach for Remote Laboratory Development," *International Journal of Online Engineering (iJOE)*, vol. 11, no. 1, pp. 99-115, 2015.
- [83] R. Tinati, X. Wang, T. Tiropanis and W. Hall, "Building a Real-Time Web Observatory," *IEEE Internet Computing*, vol. 19, no. 6, pp. 36-45, Sep. 2015.
- [84] Microsoft. Skype. (2003). [Online]. Available: https://www.skype.com, Accessed on: Nov. 15, 2016.

- [85] Facebook. WhatsApp. [Online]. Available: https://www.whatsapp.com, Accessed on: Nov. 15, 2016.
- [86] Facebook. Facebook. [Online]. Available: https://www.facebook.com/, Accessed on: Nov. 15, 2016.
- [87] Apple Inc. *FaceTime*. [Online].Available: https://support.apple.com/es-mx/HT204380, Accessed on: Nov. 15, 2016.
- [88] Google Inc. Google Duo. [Online]. Available: https://duo.google.com/, Accessed on: Nov. 15, 2016.
- [89] Google Inc. Google Allo. [Online]. Available: https://allo.google.com/, Accessed on: Nov. 15, 2016.
- [90] Comcast. YouNow. [Online]. Available: https://www.younow.com, Accessed on: Nov. 24, 2016.
- [91] Twitter Inc. *Periscope*. [Online]. Available: https://www.periscope.tv, Accessed on: Nov. 24, 2016.
- [92] A. Alohali, K. Kunze and R. Earle, "Run with Me: Designing Storytelling Tools for Runners," *Proceedings of the 2016 ACM International Joint Conference on Pervasive* and Ubiquitous Computing Adjunct - UbiComp '16, pp. 5-8, Sep. 2016.
- [93] J. Xu, K. Takagi and S. Sakazawa, "Efficient Generation of Dancing Animation Synchronizing with Music Based on Meta Motion Graphs," *IEICE Transactions on Information and Systems*, Vols. E95-D, no. E95-D, pp. 1646-1655, Jun. 2012.
- [94] M. Uchinoumi, J. K. Tan and S. Ishikawa, "A Simple-Structured Real-Time Motion Capture System Employing Silhouette Images," *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, pp. 3094-3098, Oct. 2004.
- [95] F. Cassola *et al.*, "Online-Gym: A 3D Virtual Gymnasium Using Kinect Interaction," *Procedia Technology*, vol. 13, pp. 130-138, 2014.
- [96] F. Cassola et al., "Online-Gym: Multiuser Virtual Gymnasium Using RINIONS and Multiple Kinect Devices," 2014 6th International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES), pp. 25-28, Sep. 2014.
- [97] G. Prince, *Narratology: The Form and Functioning of Narrative*. Berlin, Germany: Mounton Publisher, 1982.
- [98] S. Onega and J. Á. García Landa, *Narratology: An Introduction*. New York, NY, USA: Longman Group Limited, 1996, pp. 1-7.
- [99] M. Bal, *Narratology: Introduction to the Theory of Narrative*, 2nd ed. Toronto, Canada: University of Toronto Press, 1997.
- [100] D. Herman, *Narrative Theory and the Cognitive Sciences*, 2nd ed. CSLI Publications, 2003, p. 2.
- [101] T. Akimoto and T. Ogata, "Macro Structure and Basic Methods in the Integrated Narrative Generation System by Introducing Narratological Knowledge," 2012 IEEE 11th Int. Conf. on Cognitive Informatics and Cognitive Computing, pp. 253-262, Aug. 2012.
- [102] M. Fludernik, *An Introduction to Narratology*. New York, NY, USA: Routledge, 2009.
- [103] N. Smeda, E. Dakich and N. Sharda, "The Effectiveness of Digital Storytelling in the Classrooms: A Comprehensive Study," *Smart Learning Environments*, vol. 1, no. 1, pp. 1-21, Dec. 2014.
- [104] S. Behmer, "Literature Review Digital Storytelling: Examining the Process with Middle School Students," *Proceedings of the Society for Information Technology & Teacher Education International Conference*, vol. 30, no. 11, pp. 822-827, 2005.
- [105] S. K. Sugathan and K. S. Kalid, "An Exploratory Study of Storytelling Approach as an Instructional Tool from Educators' Perspective," 2009 International Conference on Computer Technology and Development, vol. 2, pp. 480-483, Nov. 2009.
- [106] B. R. Robin, "The Educational Uses of Digital Storytelling," Proceedings of Society for Information Technology & Teacher Education International Conference 2006, vol. 2006, no. 1, pp. 709-716, Mar. 2006.
- [107] V. Harju, K. Viitanen and M. Vivitsou, "Digital Storytelling in Finnish Schools: A Guide Towards New Ecosystems of Learning," in *Finnish Innovations and Technologies in Schools*. Rotterdam: SensePublishers, 2014, pp. 49-56.

- [108] H. J. Pitler, "Viewing Technology Through Three Lenses," *Principal*, vol. 5, no. 38-42, p. 85, Jun. 2006.
- [109] E. Wende and G. King, "Exploring Storytelling for Relationship Building in Offshore Outsourced Projects: An Action Research Investigation," System Sciences (HICSS), 2015 48th Hawaii International Conference on, pp. 412-421, Jan. 2015.
- [110] P. Rizzo, E. Shaw and W. L. Johnson, "An Agent That Helps Children to Author Rhetorically-Structured Digital Puppet Presentations," in *Intelligent Tutoring Systems*. Berlin, Germany: Springer Berlin Heidelberg, 2002, p. 2363.
- [111] J. J. McConnell, *Computer Graphics Theory Into Practice*. Sudbury, MA, USA: Jone and Bartlett Publishers, Inc., 2006.
- [112] J. D. Eisenberg and A. Bellamy-Royds, SVG Essentials, 2nd ed. Sebastopol: O'Reilly Media, Inc., 2014.
- [113] R. Chopra, *Computer Graphics With An Introduction to Multimedia*, 1st ed. New Delhi, India: S. Chand & Company Pvt. Ltd., 2011.
- [114] J. Seidelin, HTML5 Games: Creating Fun with HTML5, CSS3 and WebGL. West Sussex, England: John Wiley and Sons, Ltd., 2014.
- [115] E. Pearlman and L. House, "Introduction to SVG," in SVG for Web Developers. Upper Saddle River, NJ, USA: Pearson Education, 2003, pp. 4-5.
- [116] K. Cagle, SVG Programming: The Graphical Web. Berkeley, CA, USA: APress Media, LLC., 2002.
- [117] J. M. Carrol, "Introduction: Toward a Multidisciplinary Science of Human-Computer Interaction," in *HCI Models, Theories, and Frameworks Toward a Multidisciplinary Science*. San Francisco, CA, USA: Morgan Kaufmann Publishers, 2003, pp. 1-2.
- [118] G. J. Kim, *Human–Computer Interaction: Fundamentals and Practice*. Boca Raton, FL, USA: Taylor & Francis Group, LLC, 2015.
- [119] A. Dix, J. Finlay, G. D. Abowd and R. Beale, *Human–Computer Interaction*, 3rd ed. Essex, England: Pearson Education, 2004.

- [120] J. J. Yanguas, Modelo de Atención a las Personas con Enfermedad de Alzheimer, Instituto de Mayores y Servicios Sociales (IMERSO), 1a ed. Madrid, España: IMERSO, 2007.
- [121] T. Magal-Royo and J. L. Giménez López, "La Interactividad Multimodal en la Sección de Lengua Extranjera de la Prueba de Acceso a la Universidad en España," *Revista de Educación*, Vols. Enero-Abril 2012, no. 357, pp. 163-176, 2012.
- [122] J. Llisterri and M. J. Machuca, Los Sistemas de Diálogo. Barcelona, España: Servei de Publicacions, 2006.
- [123] Fundación Telefónica, *Realidad Aumentada: Una Nueva Lente para Ver el Mundo*, 1a ed. Barcelona, España: Editorial Ariel, S.A., 2011.
- [124] S. Daniel and P. Filmore, "Graphical Environment Tool for Development Versus Non Graphical Development Tool," in *Advances in Communications, Computing, Networks and Security*, vol. 7. Plymouth, England: University of Plymouth, 2010, pp. 255-262.
- [125] J. Yim, K. Lee and H. Kim, "Implementation of a Preliminary Natural User Interface for Video on Demand Systems," in *Future Information Technology*, vol. 309. Berlin, Germany: Springer Berlin Heidelberg, 2014, pp. 475-480.
- [126] A. Bin Tomi and D. R. Awang Rambli, "A Conceptual Design for Augmented Reality Games Using Motion Detection as User Interface and Interaction," in *Visual Informatics: Sustaining Research and Innovations*, vol. 7067. Berlin, Germany: Springer Berlin Heidelberg, 2011, pp. 305-315.
- [127] R. Vertegaal and I. Poupyrev, "Organic User Interfaces," Communications of the ACM, vol. 51, no. 6, pp. 26-30, Jun. 2008.
- [128] D. Holman and R. Vertegaal, "Organic User Interfaces: Designing Computers in any Way, Shape, or Form," *Communications of the ACM*, vol. 51, no. 6, pp. 48-55, Jun. 2008.
- [129] E. Blevis, "Sustainability Implications of Organic User Interface Technologies: An Inki Problem," *Communications of the ACM*, vol. 51, no. 6, pp. 56-57, Jun. 2008.

- [130] D. Wigdor and D. Wixon, "The Natural User Interface," in *Brave NUI World: Designing Natural User Interfaces for Touch and Gesture*. Burlington, MA, USA: Morgan Kaufmann Publishers, 2011, pp. 9-14.
- [131] M. R. Ogiela and T. Hachaj, "Natural User Interfaces for Exploring and Modeling Medical Images and Defining Gesture Description Technology," in *Natural User Interfaces in Medical Image Analysis*, Springer International Publishing, 2015, pp. 205-279.
- [132] O. Shaer and E. Hornecker, "Tangible User Interfaces: Past, Present, and Future Directions," *Foundations and Trends in Human-Computer Interaction*, vol. 3, no. 1-2, pp. 1-137, Jan. 2009.
- [133] K. Matsui, T. Terada and S. Nishio, "User Preference Learning System for Tangible User Interfaces," *Complex, Intelligent and Software Intensive Systems, 2009. CISIS* '09. *International Conference on*, pp. 766-771, Mar. 2009.
- [134] N. Couture, G. Rivière and P. Reuter, "Tangible Interaction in Mixed Reality Systems," in *The Engineering of Mixed Reality Systems*. London, England: Springer London, 2010, pp. 101-120.
- [135] R. Steinmetz and K. Nahrstedt, "User Interfaces," in *Multimedia Applications*, Berlin, Germany: Springer Berlin Heidelberg, 2004, pp. 151-172.
- [136] M. H. Cohen, J. P. Giangola and J. Balogh, "Introduction to Voice User Interfaces," in *Voice User Interface Design*. Boston, MA, USA: Addison-Wesley, 2004, pp. 3-14.
- [137] O. Schreer, P. Eisert, P. Kauff, R. Tanger and R. Englert, "Towards Robust Intuitive Vision-Based User Interfaces," 2006 IEEE International Conference on Multimedia and Expo, pp. 69-72, Jul. 2006.
- [138] G. Iannizzotto, C. Costanzo, P. Lanzafame and F. La Rosa, "A Vision-Based User Interface for Real-Time Controlling Toy Cars," 2005 IEEE Conference on Emerging Technologies and Factory Automation, vol. 1, pp. 1009-1016, Sep. 2005.
- [139] M. Kitagawa and B. Windsor, MoCap for Artists: Workflow and Techniques for Motion Capture. Oxford, England: Elsevier, 2008, pp. 1-12.

- [140] R. Tobón, "Optical Motion Capture," in *The Mocap Book: A Practical Guide to the Art of Motion Capture*. Foris Force, 2010, pp. 13-22.
- [141] J. Shin, K.-R. Ko and S. B. Pan, "Automation of Human Body Model Data Measurement Using Kinect in Motion Capture System," *Consumer Electronics* (*ICCE*), 2015 IEEE International Conference on, pp. 88-89, 9-12 Jan. 2015.
- [142] A. Menache, Understanding Motion Capture for Computer Animation, 2nd ed. Burlington, MA, USA: Morgan Kaufmann, 2011.
- [143] Microsoft. *Kinect*. [Online].
   Available: https://developer.microsoft.com/en-us/windows/kinect, Accessed on: Jan. 10, 2017.
- [144] L. Shao, J. Han, P. Kohli and Z. Zhang, *Computer Vision and Machine Learning with RGB-D Sensors*, Cham, Germany: Springer International Publishing, 2014.
- [145] K. Litomisky, "Consumer RGB-D Cameras and their Applications," Univ. of California, Riverside, 2012.
- [146] B. John Southwell and G. Fang, "Human Object Recognition Using Colour and Depth Information from an RGB-D Kinect Sensor," *International Journal of Advanced Robotic Systems*, vol. 10, pp. 1-8, 2013.
- [147] Intel. Intel<sup>®</sup> RealSense<sup>™</sup> Developer Kits. [Online].
   Available: http://click.intel.com/realsense.html, Accessed on: Jan. 10, 2017.
- [148] ASUSTeK Computer Inc. Xtion PRO 3D Sensor ASUS Global. [Online]. Available: https://www.asus.com/3D-Sensor/Xtion\_PRO/, Accessed on: Jan. 10, 2017.
- [149] PointGrab. [Online]. Available: http://www.pointgrab.com/, Accessed on: Jan. 10, 2017.
- [150] Stereolabs. ZED Stereo Camera. [Online]. Available: https://www.stereolabs.com/, Accessed on: Jan. 10, 2017.
- [151] Orbbec 3D. Orbbec. [Online]. Available: https://orbbec3d.com/, Accessed on: Jan. 10, 2017.
- [152] Tobii AB. Tobii Eye Tracking. [Online]. Available: https://tobiigaming.com/, Accessed on: Jan. 10, 2017.

- [153] D. Jacobson, G. Brail and D. Woods, *APIs: A Strategy Guide*, 1st ed. Sebastopol, CA, USA: O'Reilly Media, Inc, 2012.
- [154] J. Stylos, "Making APIs More Usable with Improved API Designs, Documentation and Tools," Ph.D. dissertation, Carnegie Mellon Univ., Pittsburgh, PA, 2009.
- [155] Facebook. *Las API y los SDK Desarrollo de aplicaciones*. [Online]. Available: https://developers.facebook.com/docs/apis-and-sdks, Accessed on: Jan. 11, 2017.
- [156] Twitter, Inc. API Overview Twitter Developers. [Online]. Available: https://dev.twitter.com/overview/api, Accessed on: Jan. 11, 2017.
- [157] Google Developers. API de Google Maps Google Developers. [Online]. Available: https://developers.google.com/maps/?hl=es-419, Accessed on: Jan. 11, 2017.
- [158] A. Durresi and R. Jain, "RTP, RTCP, and RTSP Internet Protocols for Real- Time Multimedia Communication," in *The Industrial Information Technology Handbook*. South San Francisco, CA, USA: CRC Press LLC, 2004, pp. 398-408.
- [159] R. Sharp, *Principles of Protocol Design*, 1st ed. Berlin, Germany: Springer Berlin Heidelberg, 2008.
- [160] Javvyn Techonologies, Inc., Network Protocols, 2nd ed. Saratoga, CA, USA: 2005.
- [161] H. Craig, TCP/IP Network Administration. Sebastopol, CA, USA: O'Reilly Media, Inc., 2002.
- [162] M. M. Alani, *Guide to OSI and TCP/IP Models*. Cham, Germany: Springer International Publishing, 2014.
- [163] J. Íñigo Griera et al., Estructura de Redes de Computadores. Barcelona, España: UOC, 2008.
- [164] M. Young Rhee, Internet Security: Cryptographic Principles, Algorithms and Protocols. West Sussex, England: John Wiley & Sons Ltd., 2003.
- [165] Logic Link Control, IEEE 802.2, 1998.
- [166] J. L. Carrel, L. A. Chappell, E. Tittel and J. Pyles, *Guide to TCP/IP, Course Technology*. Boston, MA, USA: Cengage Learning, 2013.
- [167] H. Ning, Unit and Ubiquitous Internet of Things. Boca Raton, FL, USA: Taylor & Francis Group, LLC, 2013.

- [168] C.-I. Zhong, Z. Zhu and R.-g. Huang, "Study on the IOT Architecture and Gateway Technology," 2015 14th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES), pp. 196-199, Aug. 2015.
- [169] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347-2376, 2015.
- [170] S. Li, L. D. Xu and S. Zhao, "The Internet of Things: A Survey," *Information Systems Frontiers*, vol. 17, no. 2, pp. 243-259, 2015.
- [171] S. Li, T. Tryfonas and H. Li, "The Internet of Things: A Security Point of View," *Internet Research*, vol. 26, no. 2, pp. 337-359, 2016.
- [172] T. Kramp, R. Van Kranenburg and S. Lange, *Enabling Things to Talk*, Berlin, Germany: Springer Berlin Heidelberg, 2013.
- [173] IoT-A, Internet-of-Things Architecture IoT-A Project Deliverable D1.2 Initial Architectural Reference Model for IoT, 2011.
- [174] A. Sharma *et al.*, "A Secure Hybrid Cloud Enabled Architecture for Internet of Things," 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT), Dec. 2015.
- [175] T. Wolf and A. Nagurney, "A Layered Protocol Architecture for Scalable Innovation and Identification of Network Economic Synergies in the Internet of Things," 2016 IEEE First International Conference on Internet-of-Things Design and Implementation, pp. 141-151, 2016.
- [176] H. König, *Protocol Engineering*, 1st ed. Berlin, Germany: Springer Berlin Heidelberg, 2012.
- [177] A. G. Blank, TCP/IP Foundations. Alameda, CA, USA: SYBEX Inc., 2004.
- [178] U. Tos and T. Ayav, "Adaptive RTP Rate Control Method," 2011 IEEE 35th Annual Computer Software and Applications Conference Workshops, pp. 7-12, Jul. 2011.
- [179] Z. Sarker, V. Singh and C. Perkins, "An Evaluation of RTP Circuit Breaker Performance on LTE Networks," 2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 251-256, May 2014.

- [180] P. Choudhury, K. R. Prasanna Kumar, G. Athithan and S. Nandi, "A Mechanism to Recover Voice from Logged VoIP Sessions Based on RTP," 2011 IEEE 5th International Conference on Internet Multimedia Systems Architecture and Application, pp. 1-5, Dec. 2011.
- [181] S. Loreto and S. P. Romano, *Real-Time Communication with WebRTC*, 1st ed. Sebastopol, CA, USA: O'Reilly Media, Inc, 2014.
- [182] WebRTC. (2012). [Online]. Available: https://webrtc.org/, Accessed on: Feb. 5, 2016.
- [183] P. Rodríguez, J. Cerviño, I. Trajkovska and J. Salvachúa, "Advanced Videoconferencing Services Based on WebRTC," *IADIS International Conferences Web Based Communities and Social Media 2012 and Collaborative Technologies* 2012, pp. 180-184, 2012.
- [184] A. B. Johnston and D. C. Burnett, WebRTC: APIs and RTCWeb Protocols of the HTML5 Real-Time Web, 2nd ed. St. Louis, MO, USA: Digital Codex LLC, 2013.
- [185] R. Manson, *Getting Started with WebRTC*. Birmingham, England: Packt Publishing Ltd., 2013.
- [186] B. Homès, *Fundamentals of Software Testing*. London, England: ISTE Ltd., 2012.
- [187] Information Technology Software Product Quality, ISO/IEC, 2000.
- [188] S. Koirala and S. Sheikh, "Software Testing Basics," in *Software Testing Interview Questions*. Sudbury, MA, USA: Infinity Science Press, 2008, pp. 1-47.
- [189] S. Desikan and G. Ramesh, Software Testing Principles and Practices. Patparganj, India: Dorling Kindersley (India) Pvt. Ltd., 2006.
- [190] B. S. Ainapure, Software Testing and Quality Assurance, 1st ed. Pune, India: Technical Publications Pune, 2008.
- [191] K. L. James, *Software Engineering*, Patparganj, India: PHI Learning, 2009.
- [192] S. Desai and A. Srivastava, Software Testing A Practical Approach, 2nd ed. Patparganj, India: PHI Learning Private Limited, 2016.
- [193] PrimeSense. PrimeSense 3D Sensors Datasheet. [Online]. Available: http://www.i3du.gr/pdf/primesense.pdf, Accessed on: Jun. 20, 2017.

- [194] Creative Technology Ltd. Intel RealSense 3D Camera: Technical Specifications.
   [Online]. Available: http://support.creative.com/kb/ShowArticle.aspx?sid=124661, Accessed on: Jun. 20, 2017.
- [195] OpenNI. OpenNI SDK. [Online]. Available: http://www.openni.ru/, Accessed on: Jan. 19, 2017.
- [196] T. Takeuchi. *Totakke*. [Online]. Available: https://github.com/totakke/homebrewopenni, Accessed on: Dec. 1, 2016.
- [197] E. González and A. Pacheco, *Proceso de Elaboración de Una Marioneta Digital MD-*02, 1 ed. Chihuahua, México: 2016.
- [198] Monitis. *Monitis*. [Online]. Available: http://www.monitis.com/es, Accessed on: Jun. 22, 2017.
- [199] Real Academia Española. Diccionario de la Lengua Española. [Online]. Available: http://dle.rae.es/?id=XXy9QSK, Accessed on: Nov. 20, 2016.
- [200] A. Pacheco. *Live Editor Avatar Demo*. [Online].Available: http://podcast.itch.edu.mx/live-editor/, Accessed on: Jun. 27, 2017.

## **ANEXO 1**

## COMPATIBILIDAD DE LOS PRINCIPALES NAVEGADORES WEB CON SVG



APÉNDICE A

GUÍA DE INSTALACIÓN DE PRIMESENSE® Y OPENNI® EN MAC OS

## INSTITUTO TECNOLÓGICO DE CHIHUAHUA

## DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

# MANUAL

# GUÍA DE INSTALACIÓN DE PRIMESENSE® Y OPENNI® EN MAC OS

LABORATORIO DE APRENDIZAJE MÓVIL

Autores:

Ing. Elisabet González Juárez

M.C. Alberto Pacheco González



Versión 1.0 Febrero 2017

## I. DESINSTALACIÓN DE CONTROLADORES PREVIOS DE PRIMESENSE<sup>®</sup>/KINECT<sup>®</sup>

Para llevar a cabo la instalación de PrimeSense<sup>®</sup> y OpenNI<sup>®</sup> lo más adecuado es hacer una instalación limpia, esto requiere la desinstalación de todos los controladores de PrimeSense<sup>®</sup> o Kinect<sup>®</sup> presentes en el equipo de cómputo.

### 1.1 Desinstalación de XQuartz<sup>®</sup>

A. Presionar la tecla cmd y la barra espaciadora para abrir la función búsqueda en *Spotlight* de macOS (Figura 1.1).



Figura 1.1. Búsqueda en Spotlight de macOS.

B. Escribir la palabra **terminal** en *Spotlight* y dar doble clic sobre el resultado con el ícono negro, ver Figura 1.2.



Figura 1.2. Resultado de búsqueda en Spotlight.

C. Cuando se haya abierto la aplicación Terminal, aparecerá una ventana como la de la Figura 1.3.

•••	👚 elisabet -	– -bash — 80×24	
Last login: Mon Feb 6 15 MacBook:~ elisabet\$	5:41:10 on ttys	5000	

Figura 1.3. Aplicación Terminal para macOS.

D. Como se muestra en la Figura 1.4, escribir en la terminal el siguiente comando:

launchctl unload -w
/Library/LaunchAgents/org.macosforge.xquartz.startx.plist

E. Presionar la tecla Return  $\leftarrow$  del teclado.



Figura 1.4. Primer comando para desinstalar XQuartz®.

F. En la terminal, escribir el siguiente comando (Figura 1.5):

Figura 1.5. Segundo comando para desinstalar XQuartz<sup>®</sup>.

G. Presionar la tecla Return  $\leftarrow$  del teclado.

H. Introducir la contraseña de la cuenta de usuario de la computadora y presionar la tecla Return *del* teclado (Figura 1.6).



Figura 1.6. Solicitud de contraseña de usuario.

I. Escribir el siguiente comando en la terminal (Figura 1.7):

sudo pkgutil --forget org.macosforge.xquartz.pkg



Figura 1.7. Tercer comando para desinstalar XQuartz®.

J. Presionar la tecla Return  $\leftarrow$  del teclado.

K. En la terminal, aparecerá el siguiente mensaje: Forgot package 'org.macosforge.xquartz.pkg' on '/'.

#### 1.2 Desinstalación de CMake

A. Repetir los pasos A, B y C de la sección anterior.

B. En la terminal, escribir el siguiente comando (Figura 1.8):

sudo rm /usr/bin/ccmake /usr/bin/cmake /usr/bin/cmake-gui
/usr/bin/cmakexbuild /usr/bin/cpack /usr/bin/ctest



Figura 1.8. Comando para desinstalar CMake.

C. Presionar la tecla Return  $\leftarrow$  del teclado.

D. Introducir la contraseña de la cuenta de usuario de la computadora y presionar la tecla Return ← del teclado.

## 1.3 Desinstalación de MacPorts

- A. Repetir los pasos A, B y C de la sección 1.1.
- B. Escribir el siguiente comando en la terminal (Figura 1.9):

```
sudo port -fp uninstall installed
```



Figura 1.9. Primer comando para desinstalar MacPorts.

C. Presionar la tecla Return  $\leftarrow$  del teclado.

D. Introducir la contraseña de la cuenta de usuario de la computadora y presionar la tecla Return del teclado.

E. Aparecerá el siguiente mensaje en la terminal: sudo: port: command not found.

F. En la terminal, escribir el siguiente comando (Figura 1.10):



Figura 1.10. Segundo comando para desinstalar MacPorts.

G. Presionar la tecla Return  $\leftarrow$  del teclado.

H. Introducir la contraseña de la cuenta de usuario de la computadora y presionar la tecla Return *del* teclado.

## **II. INSTALACIÓN Y ACTUALIZACIÓN DE XCODE®**

#### 2.1 Instalación de Xcode®

A. Abrir Spotlight, ver paso A de la sección 1.1.

B. Escribir la palabra Xcode<sup>®</sup>, si aparece un resultado en la carpeta Aplicaciones quiere decir que ya se tiene instalado Xcode<sup>®</sup> (Figura 2.1). Si ya se tiene instalada esta aplicación ir a la sección 2.3, en caso contrario continuar con el paso C.



Figura 2.1. Resultado de la búsqueda en Spotlight.

C. Abrir la App Store, para ello buscar el ícono de la Figura 2.2 y dar doble clic sobre él.



Figura 2.2. Ícono de la App Store.

D. A continuación se abrirá la página principal de la App Store, en el recuadro superior de la derecha escribir la palabra **xcode** y presionar la tecla Return  $\leftarrow$  del teclado, ver Figura 2.3.



Figura 2.3. Página de inicio de la App Store.

E. Aparecerán los resultados de la búsqueda, dar clic sobre el ícono de Xcode<sup>®</sup>, ver Figura 2.4.



Figura 2.4. Resultados de búsqueda para Xcode®.

Xcode

F. Para descargar e instalar dar clic sobre el botón Instalar (Figura 2.5).





G. A continuación comenzará la descarga de Xcode<sup>®</sup>, el avance de la descarga e instalación se mostrará en el *launchpad* o en la App Store (Figura 2.6).



Figura 2.6. Progreso de descarga de Xcode®: a) *launchpad*, b) App Store.

H. Una vez instalado, abrir Xcode® dando clic sobre su ícono (Figura 2.7).



Figura 2.7. Ícono de Xcode®.

I. Aparecerá un cuadro de diálogo preguntando si se desean instalar los componentes adicionales requeridos. Dar clic en el botón Install (Figura 2.8).



Figura 2.8. Cuadro de diálogo de instalación de componentes adicionales.

J. Escribir la contraseña de usuario de la computadora cuando el proceso de instalación así lo solicite, ver Figura 2.9. Dar clic sobre el botón OK.

$\cap$	Xcode quiere	e realizar cambios.
<b>V</b>	Ingresa tu conti	raseña para permitir esta operación.
	Usuario:	Elisabet
	Contraseña:	•••••
		Cancelar OK

Figura 2.9. Autorización para que Xcode® haga cambios en el equipo.

K. A continuación se empezarán a instalar los componentes adicionales de Xcode<sup>®</sup> (Figura 2.10).



Figura 2.10. Progreso de instalación de componentes adicionales de Xcode®.



L. Al finalizar la instalación se mostrará la pantalla de inicio de Xcode<sup>®</sup> (Figura 2.11).

Figura 2.11. Pantalla de inicio de Xcode®.

## 2.2 Instalación de Command Line Tools

A. En el menú de Xcode<sup>®</sup>, ubicado en la parte superior de la pantalla, buscar la opción More Developer Tools y dar clic sobre ésta; ver Figura 2.12.



Figura 2.12. Opción para ir al sitio de descargas para desarrolladores.

B. En el sitio de descargas, introducir los datos que se solicitan y dar clic en el botón Sign In (Figura 2.13).

Apple Inc.	Ċ
🗯 Developer	
Apple ID	
Password Create Apple ID Sign In	
Forgot ID or Password?	

Figura 2.13. Acceso al sitio de descargas para desarrolladores.

C. Buscar el instalador de *Command Line Tools* para el sistema operativo y versión de Xcode<sup>®</sup> correspondientes a los que se tengan instalados en el equipo. Descargar el archivo con extensión .dmg.

	Downloads for Apple Developers	Hi, Elisabet Gonzalez J. 👻
	Description	Release Date 🔻
+	Command Line Tools (macOS 10.12) for Xcode 8.3 beta 2	Feb 5, 2017
+	Command Line Tools (macOS 10.12) for Xcode 8.3 beta	Jan 23, 2017
+	Additional Tools <mark>for Xcode</mark> 8.2	Dec 12, 2016
	Command Line Tools (macOS 10.12) for Xcode 8.2	Dec 12, 2016
	This package enables UNIX-style development via Terminal by installing command line developer tools, as well as macOS SDK frameworks and headers. Many useful tools are included, such as the Apple LLVM compiler, linker, and Make. If you use Xcode, these tools are also embedded within the Xcode IDE.	Command Line Tools (macOS 10.12) for Xcode 8.2.dmg 147.6 MB

Figura 2.14. Sitio de descardas para desarrolladores de Apple®.

D. Abrir el instalador dando doble clic sobre éste (Figura 2.15).



Figura 2.15. Instalador para Command Line Tools.

E. A continuación se mostrarán los pasos de instalación. En la pestaña de Introducción dar clic sobre el botón Continuar, ver Figura 2.16.

😑 😑 💿 😺 🥪 Instala	r Command Line Tools (macOS Sierra version 10.12)	
	Instalador de Command Line Tools (macOS Sierra version 10.12)	
Introducción	Este programa te guiará a través de los pasos necesarios para instalar este software.	
Licencia		
Selección de     destino		
Tipo de instalación		
Instalación		
Resumen		
	Atrás Continuar	כ

Figura 2.16. Pestaña Introducción del instalador de Command Line Tools.

F. En la pestaña de Licencia dar clic sobre el botón Continuar, ver Figura 2.17.

😑 😑 💿 🛛 🤿 Instala	r Command Line Tools (macOS Sierra version 10.12)	
	Contrato de licencia	
Introducción	Español	
Licencia Selección de destino Tipo de instalación Instalación Resumen	APPLE INC. CONTRATO PARA KIT DE DESARROLLO DE SOFTWARE DE MACOS Y XCODE ROGAMOS LEA DETENIDAMENTE EL PRESENTE CONTRATO PARA EL KIT DE DESARROLLO DE SOFTWARE DE MACOS Y XCODE (EN LO SUCESIVO DENOMINADO "LICENCIA") ANTES DE UTILIZAR EL SOFTWARE DE DESARROLLADOR (QUE SE DEFINE MÁS ADELANTE). LA UTILIZACIÓN DEL SOFTWARE DE DESARROLLADOR SE INTERPRETARÁ COMO UN HECHO INEQUÍVOCO DE QUE ACEPTA LOS TÉRMINOS Y CONDICIONES DE ESTA LICENCIA. SI ACCEDE AL SOFTWARE DE DESARROLLADOR DE FORMA ELECTRÓNICA, HAGA CLIC EN EL BOTÓN "ACEPTO" PARA MANIFESTAR SU ACEPTACIÓN DE LOS TÉRMINOS Y CONDICIONES DE ESTA LICENCIA. SI NO ACEPTA DICHAS CONDICIONES, NO HAGA USO DEL SOFTWARE DE DESARROLLADOR Y HAGA CLIC EN "NO ACEPTO". NOTA IMPORTANTE: En la medida en que este software puede utilizarse para reproducir materiales, se le concede a usted bajo licencia únicamente para la reproducir de materiales sin derechos de autor, materiales de los que usted sea el propietario del copyright o materiales que esté legalmente autorizado a reproducir. En caso de desconocer si tiene derecho a reproducir o no ciertos materiales, consulte con un asesor jurídico.	
+	1. General. Imprimir Guardar Atrás Continuar	כ

Figura 2.17. Pestaña Licencia del instalador de Command Line Tools.

G. Enseguida aparecerá un recuadro solicitando la aceptación de las condiciones del contrato de licencia de software. Dar clic en el botón Acepto, ver Figura 2.18.



Figura 2.18. Aceptación de las condiciones del contrato de licencia.

H. En la pestaña Tipo de instalación dar clic en el botón Instalar (Figura 2.19).



Figura 2.19. Pestaña Tipo de instalación del instalador de Command Line Tools.

I. Introducir los datos de la cuenta del equipo (Figura 2.20) y dar clic en el botón Instalar software.

Instalador es nuevo. Ingresa tu conti	<b>stá intentando instalar un software</b> raseña para permitir esta operación.
Usuario:	Elisabet
Contraseña:	•••••
	Cancelar Instalar software

Figura 2.20. Datos para autorizar la instalación de Command Line Tools.

J. Como se muestra en la Figura 2.21, la instalación de *Command Line Tools* comenzará a realizarse.



Figura 2.21. Progreso de instalación de Command Line Tools.

K. Una vez que se haya completado la instalación se mostrará un mensaje, tal y como se muestra en la Figura 2.22. Dar clic en el botón Cerrar.



Figura 2.22. Mensaje de instalación finalizada.

## 2.3 Actualización de Xcode®

A. Para conocer si existe una actualización para Xcode<sup>®</sup> basta con abrir la App Store y dar clic sobre el ícono de actualizaciones (Figura 2.23).



Figura 2.23. Ícono de actualizaciones de la App Store.

B. En la sección de actualizaciones se mostrará si existe una nueva versión para alguna de las aplicaciones que se tengan instaladas en el equipo, entre ellas Xcode<sup>®</sup>.

## **III. INSTALACION DE SOFTWARE ADICIONAL**

#### 3.1 Descarga de XQuartz<sup>®</sup>

A. Ir al sitio <u>www.xquartz.org</u> (Figura 3.1).

••• <>		🗎 xquartz.org		C	• • • +
		XQua	rtz		
Home	The XQuartz project is an open-sou supporting libraries and application:	rce effort to develop a s, it forms the X11.app	version of the <u>X.Org X</u> that Apple shipped wit	Window System that run th OS X versions 10.5 thr	s on OS X. Together with ough 10.7.
Releases					
Support	Quick Download				
	Download	Version	Released	Info	
Contributing	XQuartz-2.7.11.dmg	2.7.11	2016-10-29	For OS X 10.6.3 or	- later
Bug Reporting	License Info				
GitHub			of cofficient which have		Due activities activities
	licenses are discussed on the <u>X.Org</u> distribution uses the <u>Apple Public S</u>	Foundation Licenses p ource License Version 2	age. The <u>guartz-wm</u> w 2.	indow manager included	with the XQuartz
	Web W	eb site content distribution services pr Distributed by Distributed by JFrog Bint	y for the XQuartz project. ovided by <u>CloudFlare</u> . y <b>ray</b>		

Figura 3.1. Sitio web de XQuartz<sup>®</sup>.

B. Para realizar la descarga, dar clic sobre el enlace debajo de la sección Download, ver Figura 3.2.

## **Quick Download**

Download	Versi	on Released	Info
<u>XQuartz-2.7.11.dmg</u>	2.7.1	1 2016-10-29	For OS X 10.6.3 or later

Figura 3.2. Enlace de descarda de XQuartz<sup>®</sup> para OS X.

C. Abrir el instalador dando doble clic sobre el ícono de la Figura 3.3.

🔴 🔴 🕒 XQuartz-2.7.11
<del>v</del>
XQuartz.pkg
UV XQuartz-2.7.11

Figura 3.3. Instalador de XQuartz<sup>®</sup>.

D. A continuación se mostrarán los pasos de instalación. En la pestaña de Introducción dar clic sobre el botón Continuar, ver Figura 3.4.

	🥪 Instalar XQuartz 2.7.11	
	Instalador de XQuartz 2.7.11	
Introducción	This is a community-supported version of the X11 windowing system for Mac OS-X 10.6.3 or later. Please visit http://www.xquartz.org for more information	
Léeme	inomator.	
Licencia		
<ul> <li>Selección de destino</li> </ul>		
<ul> <li>Tipo de instalación</li> </ul>		
Instalación		
Resumen		
	Atrás	כ

Figura 3.4. Pestaña Introducción del instalador de XQuartz®.

E. En la pestaña Léeme dar clic sobre el botón Continuar, ver Figura 3.5.



Figura 3.5. Pestaña Léeme del instalador de XQuartz<sup>®</sup>.

F. En la pestaña Licencia dar clic en el botón Continuar, ver Figura 3.6.



Figura 3.6. Pestaña Licencia del instalador de XQuartz®.

G. Enseguida aparecerá un recuadro solicitando la aceptación de las condiciones del contrato de licencia de software. Dar clic en el botón Acepto, ver Figura 3.7.



Figura 3.7. Aceptación de las condiciones del contrato de licencia.

H. En la pestaña Tipo de instalación dar clic en el botón Instalar (Figura 3.8).



Figura 3.8. Pestaña Tipo de instalación del instalador de XQuartz®.

I. Introducir los datos de la cuenta del equipo (Figura 3.9) y dar clic en el botón Instalar software.

Instalador está intentando in nuevo. Ingresa tu contraseña para permitir	<b>stalar un software</b> esta operación.
Usuario: Elisabet	
Contraseña: •••••••	
Cancelar	Instalar software

Figura 3.9. Datos para autorizar la instalación de XQuartz®.

J. Como se muestra en la Figura 3.10, la instalación de XQuartz<sup>®</sup> comenzará a realizarse.

0 0 0	🥪 Instalar XQuartz 2.7.11
	Instalando XQuartz 2.7.11
<ul> <li>Introducción</li> <li>Léeme</li> <li>Licencia</li> <li>Selección de destino</li> <li>Tipo de instalación</li> <li>Instalación</li> <li>Resumen</li> </ul>	Validando paquetes
	Atrás Continuar

Figura 3.10. Progreso de instalación de XQuartz®.

K. Una vez que se haya completado la instalación se mostrará un mensaje, tal y como se muestra en la Figura 3.11. Dar clic en el botón OK.



Figura 3.11. Indicación de reinicio de la sesión para utilizar XQuartz®.

L. Finalmente se mostrará un mensaje indicando que la instalación ha terminado, ver Figura 3.12. Dar clic en el botón Cerrar.



Figura 3.12. Mensaje de instalación finalizada.

#### 3.2 Instalación de CMake

A. Ir al sitio www.cmake.org (Figura 3.13).



Figura 3.13. Sitio web de CMake.

B. En el menú superior, buscar la opción Download y dar clic sobre ésta (Figura 3.14).



Figura 3.14. Menú del sitio web de CMake.

C. En el sitio de descargas, buscar la sección Latest Release. En la opción Binary distributions seleccionar el archivo con la extensión .dmg para la plataforma Mac OSX y dar clic para descargar, ver Figura 3.15.

#### Latest Release (3.7.2)

The release was packaged with CPack which is included as part of the release. The .sh files are self extracting gziped tar files. To install a .sh file, run it v /bin/sh and follow the directions. The OS-machine.tar.gz files are gziped tar files of the install tree. The OS-machine.tar.Z files are compressed tar files install tree. The tar file distributions can be untared in any directory. They are prefixed by the version of CMake. For example, the Linux-x86\_64 tar file i under the directory cmake–Linux-x86\_64. This prefix can be removed as long as the share, bin, man and doc directories are moved relative to each oth build the source distributions, unpack them with zip or tar and follow the instructions in Readme.txt at the top of the source tree. See also the CMake : Release Notes. Source distributions:

Platform	Files
Unix/Linux Source (has \n line feeds)	cmake-3.7.2.tar.gz
	cmake-3.7.2.tar.Z
Windows Source (has \r\n line feeds)	cmake-3.7.2.zip

Binary distributions:

Platform	Files
Windows win64-x64 Installer: Installer tool has changed. Uninstall CMake 3.4 or lower first!	cmake-3.7.2-win64-x64.msi
Windows win64-x64 ZIP	cmake-3.7.2-win64-x64.zip
Windows win32-x86 Installer: Installer tool has changed. Uninstall CMake 3.4 or lower first!	cmake-3.7.2-win32-x86.msi
Windows win32-x86 ZIP	cmake-3.7.2-win32-x86.zip
Mac OSX 10.6 or later	cmake-3.7.2-Darwin-x86_64.dmg

Figura 3.15. Sitio de descargas de CMake.

D. Buscar el archivo de instalación en la carpeta donde se haya guardado y ejecutarlo dando doble clic sobre el archivo.

E. Cuando se ejecute el instalador aparecerá un cuadro de diálogo (Figura 3.16) mostrando los términos del acuerdo de licencia, para aceptar y continuar con la instalación dar clic sobre el botón Agree.



Figura 3.16. Condiciones del contrato de licencia de CMake.

F. Una vez que se hayan aceptado los términos del acuerdo de licencia aparecerán las instrucciones de instalación. Para esto solo basta con arrastrar el ícono de CMake a la carpeta de Aplicaciones, tal como se indica en la Figura 3.17.

cmake-3.7.2-1	Darwin-x86_64
CMake	Applications
Cmake-3.7.2-Darwin-x86_64	

Figura 3.17. Instrucciones de instalación de CMake.
### **3.3 Descarga de MacPorts**

### A. Ir al sitio <u>www.macports.org</u> (Figura 3.18).

		a macports.org	Ç
Mac Ports			Download Download
Getting Started	The MacPorts Proje	ect Official Homepage	
Installing MacPorts Available Ports Documentation Support & Development Contact Us News Shortcuts Available Downloads MacPorts FAQ Report a Bug Bug reporting Guidelines Git Repository MacPorts Team Becoming a Member	The MacPorts Project is an open-source commupgrading either command-line, X11 or Aqua b command-line driven MacPorts software packa that greatly simplify the task of compiling and it we provide a single software tree that attempts them into "stable" Vs. "unstable" branches, targ. There are currently 22280 ports in our tree, dis <b>Getting started</b> For information on installing MacPorts please a provide and our base system requirements. If you run into any problems installing and/or u get get in touch with us. Other important help rowiki server & bug tracker. Latest MacPorts release: 2.4.0	nunity initiative to design an easy-to-use system for compiling, ins based open-source software on the <u>Mac operating system</u> . To that age under a <u>BSD 3-Clause License</u> , and through it easy access to <u>nstalling</u> open-source software on your Mac. Is to track the latest release of every software title (port) we distrib geting mainly OS X Mavericks v10.9 and later (including macOS stributed among 86 different categories, and more are being adde see the <u>installation</u> section of this site and explore the myriad of d sing MacPorts we also have many options to help you, depending esources are our online documentation, A.K.A <u>The MacPorts Gui</u>	talling, and t end we provide the t thousands of ports ute, without splitting Sierra v10.12). d on a regular basis. wonload options we g on how you wish to de, and our Trac
	Getting involved		
	There are many ways you can get involved wit to the "Contact Us" section of our site and:	h MacPorts and peer users, system administrators & developers	alike. Browse over

Figura 3.18. Sitio web de MacPorts.

B. En el menú izquierdo Getting Started seleccionar la opción Installing MacPorts (Figura 3.19).



Figura 3.19. Menú Getting Started del sitio web de MacPorts.

C. De acuerdo al sistema operativo que se tenga instalado en el equipo seleccionar el instalador correspondiente, ver Figura 3.20.



Figura 3.20. Sitio de descarga de MacPorts.

D. Buscar el archivo de instalación en la carpeta donde se haya guardado y ejecutarlo dando doble clic sobre el archivo.

E. A continuación se mostrarán los pasos de instalación. En la pestaña de Introducción dar clic sobre el botón Continuar, ver Figura 3.21.



Figura 3.21. Pestaña Introducción del instalador de MacPorts.

F. En la pestaña Léeme dar clic sobre el botón Continuar, ver Figura 3.22.



Figura 3.22. Pestaña Léeme del instalador de MacPorts.

G. En la pestaña Licencia dar clic en el botón Continuar, ver Figura 3.23.

• • •	🥪 Instalar MacPorts 🛛 🔒			
	Contrato de licencia			
Introducción	English			
<ul> <li>Léeme</li> <li>Licencia</li> </ul>	Copyright © 2002–2003, Apple Inc. Copyright © 2004–2016, The MacPorts Project. All rights reserved.			
<ul> <li>Selección de destino</li> <li>Tipo de instalación</li> <li>Instalación</li> </ul>	<ul> <li>Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:</li> <li>1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.</li> <li>2. Redistributions in binary form must reproduce the above copyright</li> </ul>			
<ul> <li>Resumen</li> <li>This package was made with:</li> </ul>	<ul> <li>notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.</li> <li>3. Neither the name of Apple Inc., The MacPorts Project nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.</li> </ul>			
Mac Ports http://www.macports.org/	THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND       Imprimir     Guardar       Atrás     Continuar			

Figura 2.23. Pestaña Licencia del instalador de MacPorts.

H. Enseguida aparecerá un recuadro solicitando la aceptación de las condiciones del contrato de licencia de software. Dar clic en el botón Acepto, ver Figura 3.24.

🔘 😑 😌 😜 İnstalar MacPorts					
Para continuar con la instalación, debes aceptar las condiciones del contrato de licencia de software.					
<ul> <li>Intr</li> <li>Lée</li> </ul>	Intr Haz clic en "Acepto" para continuar o en "No acepto" para cancelar la     Lée instalación y salir del Instalador.				
• Lic					
<ul> <li>Sel des</li> </ul>	Leer licencia No acepto Acepto				
<ul> <li>Tipo de</li> <li>Instala</li> <li>Resum</li> </ul>	<ol> <li>Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.</li> <li>Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.</li> <li>Neither the name of Apple Inc., The MacPorts Project nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.</li> </ol>				
Mac	Ports       THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND         .macports.org/       Imprimir       Guardar         Atrás       Continuar				

Figura 3.24. Aceptación de las condiciones del contrato de licencia.

I. En la pestaña Selección de destino dar clic en el botón Continuar (Figura 3.25).



Figura 3.25. Pestaña Selección de destino del instalador de MacPorts.

J. En la pestaña Tipo de instalación dar clic en el botón Instalar (Figura 3.26).



Figura 3.26. Pestaña Tipo de instalación del instalador de MacPorts.

K. Introducir los datos de la cuenta del equipo (Figura 3.27) y dar clic en el botón Instalar software.

$\bigcirc$	Instalador es nuevo.	stá intentando instalar un software
	Ingresa tu cont	raseña para permitir esta operación.
	Usuario:	Elisabet
	Contraseña:	•••••
		Cancelar Instalar software

Figura 3.27. Datos para autorizar la instalación de MacPorts.

L. Como se muestra en la Figura 3.28, la instalación de MacPorts comenzará a realizarse.

0 0 0	💝 Instalar MacPorts	
	Instalando MacPorts	
<ul> <li>Introducción</li> <li>Léeme</li> <li>Licencia</li> <li>Selección de destino</li> <li>Tipo de instalación</li> <li>Instalación</li> <li>Resumen</li> </ul>	Validando paquetes	
This package was made with: Mac Ports http://www.macports.org/	Atrás Continuar	

Figura 3.28. Progreso de instalación de MacPorts.

M. Finalmente se mostrará un mensaje indicando que la instalación ha terminado, ver Figura 3.29. Dar clic en el botón Cerrar.



Figura 3.29. Mensaje de instalación finalizada.

### IV. INSTALACIÓN DE CONTROLADORES DE OPENNI®

### 4.1 Instalación de Homebrew

A. Abrir la terminal, para ello repetir los pasos A, B y C de la sección 1.1.

B. Escribir en la terminal el siguiente comando (Figura 4.1):





Figura 4.1. Comando para la instalación de Hombrew.

C. Presionar la tecla Return  $\leftarrow$  del teclado.

D. Presionar nuevamente la tecla Return  $\leftarrow$  del teclado para continuar.

E. Introducir la contraseña de la cuenta de usuario de la computadora y presionar la tecla Return 
del teclado (Figura 4.2).

Figura 4.2. Solicitud de contraseña de usuario.

F. Cuando se haya completado la instalación se desplegará un mensaje en la terminal, ver Figura 4.3.

🔍 🔍 💼 💼 🕒 💼 🕒	bash — 80×24
coin	pdksh
cpp-netlib	polygen
czmqpp	pyqt
dmtx-utils	qt
dynamodb-local	qtplay
<pre>homebrew/science/bioformats-cpp</pre>	swish-e
homebrew/science/minc	tevent
homebrew/science/qgis	tlassemble
homebrew/science/sspace-longread	ttylog
libbson	tutum
libechonest	unfs3
libnice	wyrd
libqglviewer	yelp-xsl
<pre>=&gt; Installation successful!</pre>	
<pre>==&gt; Homebrew has enabled anonymous aggre Read the analytics documentation (and ho     <u>https://git.io/brew-analytics</u></pre>	gate user behaviour analytics. w to opt-out) here:
<pre>==&gt; Next steps: - Run `brew help` to get started - Further documentation: <u>https://git.io/brew-docs</u> MacBook:~ elisabet\$</pre>	



### 4.2 Instalación de *libtool*

- A. Abrir la terminal, para ello repetir los pasos A, B y C de la sección 1.1.
- B. Escribir en la terminal el siguiente comando (Figura 4.4):



Figura 4.4. Comando para instalar libtool.

- C. Presionar la tecla Return  $\leftarrow$  del teclado.
- D. Cuando se haya completado la instalación se mostrará algo similar a la Figura 4.5.



Figura 4.5. Instalación de *libtool* finalizada.

### 4.3 Instalación de libusb

- A. Abrir la terminal, para ello repetir los pasos A, B y C de la sección 1.1.
- B. Escribir en la terminal el siguiente comando (Figura 4.6):



Figura 4.6. Comando para instalar *libusb*.

- C. Presionar la tecla Return  $\leftarrow$  del teclado.
- D. Cuando se haya completado la instalación se mostrará algo similar a la Figura 4.7.





### 4.4 Instalación de homebrew-science y homebrew-openni

A. Abrir la terminal, para ello repetir los pasos A, B y C de la sección 1.1.

B. Por medio de la terminal acceder al siguiente directorio:

```
cd /usr/local
```

C. Estando en ese directorio, escribir en la terminal el siguiente comando (Figura 4.8):



D. Presionar la tecla Return  $\leftarrow$  del teclado.

E. Cuando se haya completado la instalación se mostrará algo similar a la Figura 4.9.



Figura 4.9. Instalación de hombrew-science finalizada.

F. Escribir en la terminal el siguiente comando (Figura 4.10):



Figura 4.10. Comando para instalar hombrew-openni.

- G. Presionar la tecla Return  $\leftarrow$  del teclado.
- H. Cuando se haya completado la instalación se mostrará algo similar a la Figura 4.11.



Figura 4.11. Instalación de homebrew-openni finalizada.

### 4.5 Instalación de OpenNI<sup>®</sup>, Sensor/SensorKinect y NiTE<sup>®</sup>

A. Abrir la terminal, para ello repetir los pasos A, B y C de la sección 1.1.

B. Escribir en la terminal el siguiente comando (Figura 4.12):



Figura 4.12. Comando para instalar OpenNI<sup>®</sup>.

C. Cuando se haya completado la instalación se mostrará algo similar a la Figura 4.13.

● ● ● ■ local — -bash — 80×24
==> Installing openni from homebrew/science
==> Installing dependencies for homebrew/science/openni: doxygen
==> Installing homebrew/science/openni dependency: doxygen
==> Downloading https://homebrew.bintray.com/bottles/doxygen-1.8.13.sierra.bottl
######################################
==> Pouring doxygen-1.8.13.sierra.bottle.tar.gz
<pre>/usr/local/Cellar/doxygen/1.8.13: 9 files, 13.3M</pre>
==> Installing homebrew/science/openni
<pre>==&gt; Downloading https://github.com/0penNI/0penNI/archive/Stable-1.5.7.10.tar.gz</pre>
==> Downloading from https://codeload.github.com/OpenNI/OpenNI/tar.gz/Stable-1.5
#######################################
==> Downloading https://github.com/OpenNI/OpenNI/pull/92.diff
==> Downloading from https://patch-diff.githubusercontent.com/raw/OpenNI/OpenNI/
######################################
==> Patching
==> AppLying 92.dift
patching file Samples/Niviewer/gth/gth_convenience.n
==> ./RedistMaker
==> /usr/local/cellar/openni/1.5.7.10/bin/nikeg /usr/local/cellar/openni/1.5.7.1
==> /usr/local/cellar/openn1/1.5./.10/bin/nikeg /usr/local/cellar/openn1/1.5./.1
==> /USr/local/Cellar/openni/1.5./.10/Din/nikeg/USr/local/Cellar/openni/1.5./.1
/usr/tocat/cettar/openn1/1.5./.10: 1,60/ files, 2/.9M, built in 3 minutes 42
seconds
MacBook: Local elisabets

Figura 4.13. Instalación de OpenNI® finalizada.

D. Escribir en la terminal el siguiente comando (Figura 4.14):

```
brew install sensor
```

● ● ●
MacBook:local elisabet\$ brew install sensor

Figura 4.14. Comando para instalar Sensor.

- E. Presionar la tecla Return  $\leftarrow$  del teclado.
- F. Cuando se haya completado la instalación se mostrará algo similar a la Figura 4.15.



Figura 4.15. Instalación de Sensor finalizada.

G. Escribir en la terminar el siguiente comando (Figura 4.16):

	brew install nite	
● ● ● MacBook:local eli	<pre>local — -bash — 80×24 sabet\$ brew install nite</pre>	
	Figura 4 16. Comando para instalar NiTF®	

H. Presionar la tecla Return  $\leftarrow$  del teclado.

I. Cuando se haya completado la instalación se mostrará algo similar a la Figura 4.17.



Figura 4.17. Instalación de NiTE® finalizada.

J. Ya que se ha hecho la instalación se puede ejecutar algún programa de ejemplo para comprobar el funcionamiento del sensor.

- K. Conectar el sensor al equipo.
- L. En la terminal ir al siguiente directorio:

```
cd
/usr/local/Cellar/openni/1.5.7.10/share/openni/samples/Bin/x64-
Release
```

M. Escribir el siguiente comando en la terminal (Figura 4.18):

```
./Sample-NiSimpleViewer
```

Figura 4.18. Comando para ejecutar Sample-NiSimpleViewer.

N. Al ejecutar aparecerá en la pantalla el mapa de profundidad generado por el sensor, ver Figura 4.19.



Figura 4.19. Programa de ejemplo Sample-NiSimpleViewer.

O. Si los pasos se efectuaron correctamente hasta este punto significa que el sensor PrimeSense se ha instalado correctamente.

# APÉNDICE B

### MANUAL

## PROCESO DE ELABORACIÓN DE UNA MARIONETA DIGITAL

# INSTITUTO TECNOLÓGICO DE CHIHUAHUA

# DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

# MANUAL

# PROCESO DE ELABORACIÓN DE UNA MARIONETA DIGITAL

**PROCEDIMIENTO MD-01** 

# LABORATORIO DE APRENDIZAJE MÓVIL

Autores:

Ing. Elisabet González Juárez

Gabriela Lizeth Sandoval Ordoñez

M.C. Alberto Pacheco González



Versión 1.17 Noviembre 2016

### I. DESCARGA E INSTALACIÓN DE INKSCAPE®

Para realizar la vectorización de la imagen se utiliza el software Inkscape<sup>®</sup>, el cual es un editor profesional de gráficos vectoriales para Windows<sup>®</sup>, Mac OS X<sup>®</sup> y Linux<sup>®</sup>. Es un software libre y de código abierto que permite convertir una imagen de mapa de bits a un conjunto de vectores.

### 1.1 Instalación de Inkscape®

A. Descargar el instalador del sitio <u>www.inkscape.org/es/</u>, para ello ir a la sección de descargas, ver Figura 1.1.



Figura 1.1. Sitio www.inkscape.org/es/.

B. Elegir el sistema operativo en el cual se instalará Inkscape®, ver Figura 1.2.



Figura 1.2. Sección de descargas del sitio web de Inkscape®.

### 1.1.1 Mac OS $X^{\ensuremath{\mathbb{R}}}$

A. Seleccionar el instalador para Mac OS X<sup>®</sup>, ver Figura 1.3.

Consigue la versión estable para tu sistema



Figura 1.3. Enlace de descarga para Mac OS X®.

B. Ir al enlace del archivo .dmg y dar clic sobre él para iniciar la descarga del instalador, ver Figura 1.4.

	ACERCA DE	DESCARGAS	NOTICIAS	COMUNIDAD	APRENDE	CONTRIBUYE
Inicio » Des	cargas » MAC O	s x				
Latest s	table versi	on (0.91)				
This.dmg i	ile is for Mac OS	<b>5 X 10.7-10.10</b> ar	nd requires XQ	Quartz.		
For <b>OS X ve</b> The packag <b>0.91.1</b> will h	ersion 10.5 and e for 10.5 and 1 have support for	<b>10.6 only</b> : down 0.6 is missing su r these formats.	load this <mark>.dmg</mark> pport for Visio	file (requires XQ and WPD/WPG f	uartz). ïles due to an o	oversight by the
F	Figura 1.4. Pa	ágina de deso	carga del in	stalador para	Mac OS X®	).
ar el insta	alador en la	a carpeta d	londe se	hava desca	rgado par	a posterio

C. Buscar el instalador en la carpeta donde se haya descargado para posteriormente ejecutarlo.

D. Arrastrar el ícono de Inkscape<sup>®</sup> a la carpeta Applications como se ve en la Figura 1.5.

INKSCAPE Draw Freely.	ape
	Inkscape 0.91
	<b>\$</b> .
Applications	Inkscape
Inkscape is Free and Open Source So	oftware licensed under the GPL.

Figura 1.5. Instalador de Inkscape® para Mac OS X®.

#### E. Ir a http://xquartz.org.

F. Dar clic sobre la versión más reciente de XQuartz<sup>®</sup> para descargar el instalador, ver Figura 1.6.

# XQuartz

The XQuartz project is an open-source effort to develop a version of the X.Org X Window System that runs on OS X. Together with supporting libraries and applications, it for

#### Quick Download

Download	Ver	sion	Released
XQuartz-2.7.9.dmg	2.7	7.9	2016-05-05

#### License Info

An XQuartz installation consists of many individual pieces of software which have various licenses. The X.Org software components' licenses are discussed on the X.Org Found the Apple Public Source License Version 2.

Figura 1.6. Sitio de descarga de XQuartz<sup>®</sup>.

G. Dar doble clic sobre el ícono del instalador, ver Figura 2.7.



Figura 1.7. Ícono del instalador de XQuartz<sup>®</sup>.

H. A continuación aparecerá un cuadro de diálogo (Figura 1.8). Para continuar con la instalación dar clic sobre el botón Continuar.

	💝 Instalar XQuartz 2.7.9			
Instalador de XQuartz 2.7.9				
Introducción	This is a community-supported version of the X11 windowing system for Mac OS-X 10.6.3 or later. Please visit http://www.xquartz.org for more information			
Léame	mornauon.			
Licencia				
Destino				
Tipo instalación				
Instalación				
Resumen				
	Retroceder	J		

Figura 1.8. Cuadro de diálogo del instalador de XQuartz®.

I. Dar clic nuevamente sobre el botón Continuar, ver Figura 1.9.



Figura 1.9. Cuadro de diálogo del instalador: Léame.

J. En la etapa del contrato de licencia dar clic en el botón continuar, ver Figura 1.10.



Figura 1.10. Cuadro de diálogo del instalador: Licencia.

K. Una vez hecho esto se mostrará el cuadro de diálogo de la Figura 1.11, dar clic sobre el botón Acepto.



Figura 1.11. Mensaje de aceptación del contrato de licencia.

L. En la siguiente parte dar clic en la opción Instalar, ver Figura 1.12.



Figura 1.12. Cuadro de diálogo del instalador: Tipo instalación.

M. Si el sistema lo solicita, proporcionar un nombre de usuario y contraseña para autorizar la instalación, posteriormente dar clic en Instalar software, ver Figura 1.13.

Instalador está intentando instalar software nuevo. Introduzca su contraseña para permitir esta operación.			
Nombre de usuario:	Colaborador		
Contraseña:	•••••		

Figura 1.13. Cuadro de diálogo para autorizar la instalación de software.

N. A continuación se iniciará la instalación, ver Figura 1.14.



Figure 1.14. Proceso de instalación.

O. Una vez concluida la instalación se mostrará la ventana de la Figura 1.15. Dar clic en el botón Cerrar.

	💝 Instalar XQuartz 2.7.9	
	La instalación se ha completado correctamente.	
<ul> <li>Introducción</li> <li>Léame</li> <li>Licencia</li> <li>Destino</li> <li>Tipo instalación</li> <li>Instalación</li> <li>Resumen</li> </ul>	<b>Instalación completada.</b> El software se ha instalado.	
	Retroceder	)

Figura 1.15. Instalación finalizada.

### 1.1.2 Windows®

A. Seleccionar el instalador para Windows<sup>®</sup>, ver Figura 1.16.



Figura 1.16. Enlace de descarga para Windows®.

B. Dependiendo del tipo de sistema con el que se cuente escoger el instalador para 32 bits o 64 bits. Para iniciar la descarga dar clic sobre el ícono, ver Figura 1.17.



Figura 1.17. Página de descarga del instalador para Windows®.

C. En el cuadro de diálogo que aparece dar clic en la opción Guardar archivo. Ver Figura 1.18.



Figura 1.18. Cuadro de diálogo para guardar el instalador.

D. Buscar el instalador en la carpeta donde se haya descargado para posteriormente ejecutarlo.

E. A continuación se abrirá el cuadro de diálogo del instalador (Figura 1.19). Para iniciar con la instalación dar clic en Next.



Figura 1.19. Cuadro de diálogo del instalador de Inkscape®.

F. Enseguida verificar que la casilla de la parte inferior esté seleccionada, ya que la casilla haya sido activada dar clic en el botón Next, ver Figura 1.20.

虔	Inkscape 0.91 Setup 🚽 🗖 💌			
E	nd-User License Agreement Please read the following license agreement carefully			
	GNU GENERAL PUBLIC LICENSE			
	Version 2, June 1991			
Copyright © 1989, 1991 Free Software Foundation, Inc. 51 Exception Street Eigh Floer Destee MA 02110 1201 USA				
	Print Back Next Cancel			

Figura 1.20. Aceptación de los términos de licencia.

G. Seleccionar el tipo de instalación requerida dando clic sobre el botón correspondiente. Se recomienda realizar una instalación típica, ver Figura 1.21.

🔡 Inkscape 0.91 Setup – 🗆 🗙				
Choose Setup Type Choose the setup type that best suits your needs				
<u>Typical</u> Installs the most common program features. Recommended for most users.				
Custom Allows users to choose which program features will be installed and where they will be installed. Recommended for advanced users. Complete				
All program features will be installed. Requires the most disk space. Back Next Cancel				
Back Next Cancel				

Figura 1.21. Tipos de instalación.

H. Para comenzar la instalación dar clic en el botón Install, ver Figura 1.22.



Figura 1.22. Cuadro de diálogo del instalador.

I. A continuación, aparecerá una pantalla similar a la Figura 1.23, donde se indica que se está preparando la instalación.



Figura 1.23. Preparación de la instalación.

J. En la mayoría de las ocasiones, antes de que se inicie la operación, aparecerá un cuadro de diálogo pidiendo permisos para proceder con la instalación. Dar clic en el botón Sí, ver Figura 1.24.



Figura 1.24. Cuadro de diálogo para autorizar la instalación de software.

K. Una vez que se dieron los permisos necesarios se iniciará el proceso de instalación, tal y como se muestra en la Figura 1.25.

17	Inkscape 0.91 Setup	-	×
Ir	istalling Inkscape 0.91		٠
Ple	ease wait while the Setup Wizard installs Inkscape 0.91.		
St	atus: Updating component registration		
	<u>B</u> ack <u>N</u> ext		Cancel

Figura 1.25. Progreso de la instalación.

L. Cuando la instalación haya finalizado aparecerá el cuadro de diálogo de la Figura 1.26. Para terminar, dar clic en el botón Finish.



Figura 1.26. Instalación finalizada.

### II. TRAZADO DE LA IMAGEN

El primer paso para elaborar una marioneta digital es contar con el dibujo a trabajar, éste se puede diseñar manualmente a papel y lápiz o bien puede utilizarse una imagen digital. Ya sea que el dibujo se haga de forma manual o no, es necesario cubrir una serie de especificaciones.

En la sección 2.1 se describen los pasos a seguir para dibujar manualmente el personaje.

En la sección 2.2 se muestran las características que debe tener el dibujo si se utiliza una imagen digital.

Por último, en la sección 2.3 se brindan algunas recomendaciones generales para obtener un buen diseño del personaje.

### 2.1 Elaboración del dibujo

El dibujo debe calcarse con un lápiz color negro en una hoja de máquina tamaño carta color blanco, tal y como se muestra en la Figura 2.1.



Figura 2.1. Dibujo trazado a lápiz.

El trazo debe estar centrado dentro de los márgenes de la hoja y debe ser plasmado en una posición de firmes como se muestra en la Figura 2.2.



Figura 2.2. Marioneta en posición de firmes.

Al terminar el dibujo, se deben remarcar todos y cada uno de los bordes con un marcador permanente color negro de punta mediana (no punto fino) para establecer el contorno de las partes del personaje, ver Figura 2.3.



Figura 2.3. Dibujo remarcado.

Es de gran importancia no dejar discontinuidades en los trazos ni empalmar líneas ya que al momento de vectorizar la imagen podrían obtenerse resultados poco efectivos, en la Figura 2.4 se muestran algunos ejemplos de esto.



Figura 2.4. Errores comunes de diseño: a) trazo fuera de lugar, b) empalme de líneas, c) discontinuidad de línea.

### 2.2 Selección de la imagen

Para seleccionar una imagen adecuada es importante tener en consideración las mismas especificaciones de la sección 2.1.

La imagen digital debe ser a blanco y negro, donde los bordes sean de color negro y el fondo de color blanco, así mismo el dibujo deberá encontrarse en una posición de firmes. Es importante evitar los errores de la Figura 2.4.

El formato de la imagen digital debe ser PNG (Portable Network Graphics) o JPG.

## 2.3 Recomendaciones generales

Para conseguir un diseño óptimo en la generación del dibujo es necesario seguir las siguientes recomendaciones generales sin importar el método que se haya seleccionado para esto.

Si el dibujo cuenta con demasiados detalles será difícil trabajar con él además esto podría acrecentar el tamaño del archivo.

Otro aspecto relevante es la definición de los bordes, si estos son muy delgados es posible que existan problemas en la etapa de vectorización, por otro lado si son muy gruesos se puede perder el valor estético de la figura. Al descargar una imagen digital es importante conocer con qué tipo de licencia cuenta y el uso que se le puede dar.

## III. DIGITALIZACIÓN DE LA IMAGEN

Una vez terminado el trazado y después de haber remarcado todos los bordes, es necesario utilizar un escáner para obtener la imagen digitalizada.

El escáner provee una vista previa del dibujo mostrando las propiedades del documento en el menú del escáner.



Verificar que las opciones sean iguales a las de la Figura 3.1.

Figura 3.1. Propiedades del documento a escanear.

En caso de que la imagen de visión general en el escáner tenga algún detalle puede corregirse por medio del menú de corrección de imágenes del escáner; con esto es posible variar contraste y brillo, entre otras opciones, ver Figura 3.2. Si esto no es suficiente se deberá retocar el dibujo manualmente en la hoja, utilizando corrector blanco y marcador negro. Una vez hecho esto se repite el proceso de digitalización.
•••	Escáner			
	1	Tamaño:	Carta EE. UU.	
		Orientación:	<mark></mark> ∋• Ψ.•∈	
(ඉම		Escanear para:	imágenes ᅌ	
		Nombre:	Escanear	
A A		Formato:	PNG ᅌ	
		Correc. imágenes:	Manual	1
				Ш
				Ш
<b>V</b> V		Brillo:	* *	Ш
		Contraste:	• • • • • • • •	ш
L L L L L L L L L L L L L L L L L L L			Restaurar valores por omisión	
	-	cultar detalles	Visión general Escanear	

Figura 3.2. Menú de corrección de imágenes.

Para escanear la imagen se deberá hacer un recuadro que encierre todos los elementos que se deseen digitalizar.

#### IV. VECTORIZACIÓN DE LA IMAGEN

Una vez instalado el software Inkscape<sup>®</sup> se procede a realizar el proceso de vectorización para poder manipular la imagen y crear correctamente el personaje deseado.

#### 4.1 Importación de la imagen

Para importar correctamente la imagen se siguen los siguientes pasos:

A. Abrir el programa (Inkscape<sup>®</sup>) y en el lugar de trabajo (Figura 4.1) abrir la imagen que se digitalizó utilizando el menú de la Figura 4.2.



Figura 4.1. Área de trabajo de Inskcape<sup>®</sup>.

Archivo	Edición	Ver	Capa	Objeto	Trayecto	Text	
🗋 <u>N</u> uev	0				Ct	rl+N	
🖹 <u>T</u> emp	olates				Ctrl+A	lt+N	
📙 <u>A</u> brir					Ctr	l+0	
Abrir	<u>r</u> eciente					>	
\overline Re <u>v</u> er	rtir					ĺ	
📊 <u>G</u> uar	dar				Ct	rl+S	
🔏 G <u>u</u> ar	dar como		Shift+Ctrl+S				
Guar	dar una c	op <u>i</u> a		Sł	nift+Ctrl+A	lt+S	
- Impo	rtar				C	trl+I	
🕒 Expo	rtar imag	en PN(	G		Shift+Ctrl+E		
📇 Impri	mir				Ct	rl+P	
🛃 Limpi	iar doc <u>u</u> m	ento					
🕎 Propi	iedades d	el <u>d</u> oo	cumento	)	Shift+Ct	rl+D	
💥 <u>C</u> erra	ar				Ctr	l+W	
🐐 <u>S</u> alir					Ctr	l+Q	

Figura 4.2. Menú archivo de Inkscape<sup>®</sup>.

B. En el cuadro de diálogo, buscar la carpeta en la que se encuentre el archivo, seleccionarlo y dar clic en el botón inferior Open, ver Figura 4.3.



Figura 4.3. Cuadro de diálogo del menú Archivo de Inkscape®.

C. Verificar que las opciones de importación estén similares a las de la Figura 4.4.



Figura 4.4. Opciones de importación de la imagen.

D. Después de haber importado la imagen, ésta se mostrará en el área de trabajo, ver Figura 4.5.



Figura 4.5. Figura importada.

#### 4.2 Vectorización de la imagen

Para llevar a cabo el vectorizado de la imagen es necesario seguir los siguientes pasos:

A. Seleccionar la imagen utilizando la herramienta Seleccionar y transformar objetos

B. Una vez seleccionada la imagen, ir al menú Trayecto > Vectorizar mapa de bits, ver Figura 4.6.



Figura 4.6. Menú trayecto de Inkscape®.

C. Verificar que las opciones de importación estén similares a las de la Figura 4.7 y activar la casilla Vista en directo.



Figura 4.7. Cuadro de opciones de vectorización de mapas de bits.

D. Variar el valor del umbral del Corte de luminosidad hasta que se obtenga la calidad de deseada de la imagen, dar clic en el botón OK y cerrar el cuadro de diálogo.

E. Dar clic fuera de la línea punteada pero dentro del área de trabajo, de esta manera se seleccionará la imagen PNG que quedó detrás de la imagen vectorizada, ver Figura 4.8.



Figura 4.8. Imagen PNG.

F. Quitar dicha imagen utilizando el clic derecho del ratón y seleccionando la opción Eliminar, ver Figura 4.9.



Figura 4.9. Menú eliminar imagen.

G. Guardar la imagen con el menú Archivo > Guardar como; buscar la carpeta en donde se guardará la imagen vectorizada y seleccionar el formato SVG de Inkscape<sup>®</sup>, otorgar un nombre y dar clic en el botón Save.

			X Selec	cione el archivo en el qu	ue se guardará		
Name: Grego	rio <mark>.svg</mark>						
Save in folder: < IT	CH MCIE	Tesis	Etapa de Diseño	Marionetas Digitales	Marionetas Vectorizadas SVG	Gregorio >	Create Folder
Places Search Recently Used Inkscape.app Colaborador Desktop File System templates	Name						
✓ Añadir extensión de ✓ Activar vista prelimin	archivo aut	tomática	imente	2	iVG de Inkscape (*.svg)	Cancel	Save
						curret	

Figura 4.10. Cuadro de diálogo para guardar el archivo generado.

#### 4.3 Descombinar la imagen

A. Para separar la imagen vectorizada se debe seleccionar la imagen e ir al menúTrayecto > Descombinar. La imagen se mostrará similar a la Figura 4.11.



Figura 4.11. Imagen descombinada.

B. Activar el editor XML en el menú Edición > Editor XML y cambiar las dimensiones de la etiqueta <svg> principal (Figura 4.12), de la siguiente manera:

height: 400 width: 400

®	Editor XML (Shift+Ctrl+X)	• ×	٩
			٩
	<pre>svg:svg id="svg3406"&gt;</pre>		
	🛨 <svg:metadata id="metadata3412"></svg:metadata>		
	<svg:defs id="defs3410"></svg:defs>		
	<sodipodi:namedview <="" id="namedview3408" td=""><td></td><td>ľ</td></sodipodi:namedview>		ľ
	<svg:path id="path4219"></svg:path>		٥.
	<svg:path id="path4217"></svg:path>		_
	<svg:path id="path4215"></svg:path>		^∕₀
	<svg:path id="path4213"></svg:path>		2
	<svg:path id="path4211"></svg:path>		
	<svg:path id="path4209"></svg:path>		-10-
	<svg:path id="path4207"></svg:path>		2
	<svg:path id="path4205"></svg:path>		٦,
	<svg:path id="path4203"></svg:path>		2
	<svg:path id="path4201"></svg:path>		Hap
	<svg:path id="path4199"></svg:path>		•
	<svg:path id="path4197"></svg:path>		ે
	<svo:nathid="nath4195"></svo:nathid="nath4195">		۰

Figura 4.12. Editor XML.

C. Para aplicar los cambios dar clic en el botón Aceptar, ver Figura 4.13.

Editor XML (Shift+Ctrl+X)				•
<pre>svg:svg id="svg3406"&gt;</pre>		Nombre 🗸	Valor	
+ <svg:metadata id="metadata3412"></svg:metadata>	11	height	400	
<svg:defs id="defs3410"></svg:defs>		id	svg3406	
<sodipodi:namedview id="namedview3408"></sodipodi:namedview>	н	inkscape:version	0.91 r13725	
<svg:path id="path4219"></svg:path>	ш	sodipodi:docname	Gregorio.svg	
<svg:path id="path4217"></svg:path>	ш	version	1.1	
<svg:path id="path4215"></svg:path>	ш	viewBox	0 0 560.90912 917.27271	
<svg:path id="path4213"></svg:path>		width	400	
<svg:path id="path4211"></svg:path>	н			
<svg:path id="path4209"></svg:path>	н			
<svg:path id="path4207"></svg:path>				
<svg:path id="path4205"></svg:path>	•			
<svg:path id="path4203"></svg:path>				
<svg:path id="path4201"></svg:path>				
<svg:path id="path4199"></svg:path>				
<svg:path id="path4197"></svg:path>				
<svg:path id="path4195"></svg:path>				
<svg:path id="path4193"></svg:path>				
<svg:path id="path4191"></svg:path>				
<svg:path id="path4189"></svg:path>		hoight		Acoptar
<svg:path id="path4187"></svg:path>	ſ	100		Aceptai
<svg:path id="path4185"></svg:path>	L	400		
<svg:path id="path4183"></svg:path>				
Ha seleccionado el atributo height. Cuando termine, pulse Ctrl+Enter para gua	ırda	r los cambios.		

Figura 4.13. Cuadro de propiedades de la etiqueta SVG principal.

D. Si el gráfico tiene el atributo viewBox (Figura 4.14), éste debe ser eliminado borrando su contenido.

Editor XML (Shift+Ctrl+X)				a 🛛
<pre>svg:svg id="svg3406"&gt;</pre>		Nombre 🗸	Valor	
+ <svg:metadata id="metadata3412"></svg:metadata>		height	400	
<svg:defs id="defs3410"></svg:defs>		id	svg3406	
<sodipodi:namedview id="namedview3408"></sodipodi:namedview>		inkscape:version	0.91 r13725	
<svg:path id="path4219"></svg:path>		sodipodi:docname	Gregorio.svg	
<svg:path id="path4217"></svg:path>		version	1.1	
<svg:path id="path4215"></svg:path>	1	viewBox	0 0 560.90912 917.27271	
<svg:path id="path4213"></svg:path>	I	width	400	
<svg:path id="path4211"></svg:path>				
<svg:path id="path4209"></svg:path>				
<svg:path id="path4207"></svg:path>				
<svg:path id="path4205"></svg:path>				
<svg:path id="path4203"></svg:path>				
<svg:path id="path4201"></svg:path>				
<svg:path id="path4199"></svg:path>				
<svg:path id="path4197"></svg:path>				
<svg:path id="path4195"></svg:path>				
<svg:path id="path4193"></svg:path>				
<svg:path id="path4191"></svg:path>				
<svg:path id="path4189"></svg:path>		viewBox		Acentar
<svg:path id="path4187"></svg:path>	ſ	0.0.560.00012.017	27271	Aceptai
<svg:path id="path4185"></svg:path>	Ľ	0 0 300.90912 917	.27271	
<svg:path id="path4183"></svg:path>				
La seleccionado el atributo viewRoy. Cuando termine, pulse Ctrl - Enter para cua	arc	lar los cambios		
na seleccionado el autorito viewbox, cuando termine, puse Curi+Enter para gua	aru	a ios cambios.		

Figura 4.14. Elemento viewBox de la etiqueta principal SVG.

E. Al eliminar su valor la imagen quedará fuera del cuadro, ver Figura 4.15.



Figura 4.15. Imagen fuera de cuadro.

F. Para ajustar las dimensiones de la imagen al tamaño del área de trabajo se deben seleccionar todos los elementos independientes del gráfico, ver Figura 4.16.



Figura 4.16. Selección de cada parte de la imagen fuera del área de trabajo.

G. Una vez que todos los elementos de la imagen han sido seleccionados, modificar los valores W y H de la barra de herramientas localizada en la parte superior, ver Figura 4.17.

Para mantener la proporción de la imagen, utilizar la herramienta de bloqueo de dimensiones



Figura 4.17. Dimensiones de la imagen.

H. A la dimensión más alta asignarle un valor de 400 px, el valor de la otra dimensión se actualizará automáticamente.

I. Para centrar la imagen dentro del cuadro, utilizar la herramienta de alineación y distribución En el campo Alinear utilizar las opciones de la Figura 4.18.

₽Alinear y distribuir (Shift+Ctrl+A) ④							
Alinear							
	Relativo	a: P	ágina			~	
	Tratar a	la sel	ección	como	grupo	o: 🗹	
	g		₽		P	ÿ	
		TUY	ŧ₽₽	tur	۳ľ	ya	

Figura 4.18.Caja de herramientas Alinear y distribuir.

J. Posteriormente realizar una alineación en el eje vertical alineación en el eje horizontal de esta manera el gráfico quedará al centro del área de trabajo como se muestra en la Figura 4.19.



Figura 5.19. Figura centrada.

K. Una vez que el gráfico ha sido escalado, modificar nuevamente el atributo viewBox y asignar el valor: 0 0 400 400 y guardar el archivo.

#### V. COLOREADO DE LA IMAGEN

Una vez que hemos vectorizado nuestra imagen pasamos a darle color para distinguir todas y cada una de las partes que la integran.

A. Seleccionar la silueta de toda la figura para cambiar su color y contorno. Para ello se debe utilizar la herramienta Relleno y borde. El cuadro de diálogo aparecerá del lado derecho de la interfaz de Inkscape<sup>®</sup>, tal como se muestra en la Figura 5.1.



Figura 5.1. Cuadro de diálogo Relleno y borde.

B. Cambiar al modo rueda para visualizar mejor la paleta de colores, posteriormente elegir el color deseado dando clic sobre él, ver Figura 5.2.



Figura 5.2. Cuadro de opciones de la herramienta Relleno.

C. Ir a la pestaña Color de trazo y seleccionar la opción Color uniforme. Una vez que aparezca la rueda de colores seleccionar el color deseado dando clic sobre él.



Figura 5.3. Cuadro de opciones de la herramienta Color de trazo.

D. A continuación se procede a colorear la marioneta, para ello seleccionar las diferentes secciones del dibujo y dar el color deseado, ver Figura 5.4.



Figura 5.4 Selección de un trayecto de la marioneta.

E. Al finalizar se obtendrá algo similar a la Figura 5.5.



Figura 5.5. Marioneta coloreada.

F. Una vez que la marioneta fue coloreada en su totalidad, se debe guardar la imagen en formato SVG, con un nombre distinto para conservar una copia del original.

#### VI. SECTORIZADO DE LA IMAGEN

A. Enseguida se procederá a seccionar el trayecto de la silueta de la imagen. Para ello se hará un acercamiento sobre las áreas de interés con el procedimiento mostrado en la Figura 6.1.

Ver	Сара	Obieto	Travecto	Texto	Filtros	Extension	s	Avuda	
	Zoom					>	Ð	Acercar	+
	Modo de	e visualiza	ción			>	Q	Alejar	-
	Modo de	visualiza	ción de col	or		>	11	Zoom 1:1	1
	Page Gri	d				#	12	Zoom 1:2	2
	Guías					1	21	Zoom 2:1	
	Vista de	gestión d	e color				$\bigcirc$	Selección	3
	Mostrar/	ocultar				>	6	Dibujo	4
1%	Mostrar/	ocultar di	álogos			F12	Q	Página	5
	Muostras				sh	ift + Ctrl + W	Q	Ancho de página	6
	Mensajes	5			511	III TCUITW	0	Zoom anterior	
	Ventana	anterior			Shift	+Ctrl+Tab	0	Zoom siguiente	Shift+`
	Ventana	siquiente			Jiiit	Ctrl+Tab			
	10.1								
3	vista de Duplicar	icono							
	Duplicar	ventaria							
¢	Pantalla (	completa				F11			
	Predeter	minada							
	Personali	izada							
	Ancho								

Figura 6.1. Menú Zoom.

B. Utilizar la herramienta para dibujar a mano alzada, localizada en el menú de herramientas en la parte izquierda de la ventana.

C. Dar clic sobre la imagen para marcar los puntos de inicio y fin de la recta (Figura 6.2).



Figura 6.2. Línea de división.

D. Cambiar a la herramienta para edición de nodos de trayecto.



E. Seleccionar la recta y manipularla hasta obtener la forma deseada. Verificar que la línea sobrepase en sus extremos el contorno de la imagen (Figura 6.3).



Figura 6.3. Intersección del contorno con la línea.

F. Cambiar a la herramienta de selección y con la tecla shift del teclado seleccionar primeramente la línea dibujada y después el contorno de la imagen.

Después, ir al menú Trayecto > División. Al aplicar esta transformación se generarán dos nuevos trayectos que se pueden identificar por las líneas punteadas (Figura 6.4).

Trayecto	Texto	Filtros	Extensiones	Ayuda	
🔓 <u>O</u> bjeto	o a traye	cto	Shift	t+Ctrl+C	
🧬 <u>B</u> orde	a trayed	to	Ct	rl+Alt+C	
🖉 Vec <u>t</u> or	rizar ma	pa de bit	s Shi	ft+Alt+B	
🔁 Trace	Pixel Art				
🜈 <u>U</u> nión				Ctrl++	
C <u>D</u> ifere	ncia			Ctrl+-	
🕤 Interse	ección			Ctrl+*	
E <u>x</u> clusi	ión			Ctrl+^	
🖸 Di <u>v</u> isió	'n			Ctrl+/	
Cor <u>t</u> ar	trayecto	)	Ct	rl+Alt+/	
Combi	inar			Ctrl+K	
Desco	mbinar		Shift	t+Ctrl+K	

Figura 6.4. Herramienta División.

G. Repetir el procedimiento hasta haber seccionado cada región de interés de la silueta de la imagen (Figura 6.5).



Figura 6.5. Marioneta seccionada.

H. Una vez que la marioneta fue seccionada en su totalidad, se debe guardar la imagen en formato SVG, con un nombre distinto para conservar una copia del original.

#### VII. ASIGNACIÓN DE PIVOTES

Una vez que la imagen ha sido vectorizada es momento de mapear los pivotes a fin de que la cámara de profundidad detecte cada sección de la marioneta y ésta pueda ser animada exitosamente.

#### 7.1 Identificación de las partes del cuerpo

A. Realizar una copia de seguridad del archivo. Ir al menú File > Save as...

Buscar el directorio en donde se guardará la imagen y dar clic en el botón Save (Figura 7.1).

File Edit Code Navigation	
New ►	
Open	
Close	Save 'Gregorio_coloreado.svg' As
Recently Closed Tabs	
Recent Files	Save As: Gregorio_pivoteado.svg
Save #S	Tags:
Save As Other	Where: 📄 Gregorio ᅌ
Revert ^#U	
Save All ^#S	Format: XML Files
Refresh Status	
Print ►	Cancel Save

Figura 7.1. Menú Save As.

B. Para colocar los pivotes se debe identificar cada una de las partes del cuerpo de la marioneta, esto se puede realizar tomando como base la Figura 7.2 y Figura 7.3.



Figura 7.3. Partes de la marioneta.

#### 7.2 Creación y ubicación de pivotes correspondientes.

A. Se deberán crear 10 pivotes, para ello seleccionar la herramienta para crear círculos, elipses y arcos y dibujar un círculo.

B. Establecer un radio con valor de 3 en "rx" y en "ry" por medio del editor XML, de esta manera se generara la propiedad "r" la cual tendrá un valor de 3. (Figura 7.4)

Nombre 🗸	' Valor		rx	Aceptar	Nombre 🗸	Valor
сх	55.111996		5		сх	55.111996
cy	351.19745				cv	351,19745
id	path4138				-,	
rx	16.246889	Ŀ	ry	Aceptar	Id	path4138
ry	16.884022	3	3		r	3
style	fill:#000000;fill-opacity:1				style	fill:#000000;fill-opacity:1

Figura 7.4. Valor de rx y ry.

C. Para dibujar individualmente cada uno de los 10 pivotes seleccionar el primer círculo e ir al menú>Edición> Duplicar (Figura 7.5), con esto se obtendrá una copia del círculo



Edición Ver Capa Objeto Trayecto	Texto Filtros
👆 Deshacer: Cambiar atributo	Ctrl+Z
🕐 Rehacer	Shift+Ctrl+Z
🗬 Historia de deshacer	Shift+Ctrl+H
🖓 🔏 Cortar	Ctrl+X
🕞 Copiar	Ctrl+C
🔋 Pegar	Ctrl+V
🛱 Pegar en el sitio	Ctrl+Alt+V
🖺 Pegar estilo	Shift+Ctrl+V
Pegar tamaño	>
🔍 Buscar/Reemplazar	Ctrl+F
🛅 Duplicar	Ctrl+D
Clonar	>
💼 Crear copia en mapa de bits	Alt+B
🗑 Eliminar	Delete
🖹 Seleccionar todo	Ctrl+A
🗎 Seleccionar todo de todas las capas	Ctrl+Alt+A
Seleccionar mismo	>

Figura 7.5. Herramienta Duplicar.

D. Una vez que se haya duplicado el círculo éste aparecerá exactamente en la misma posición que el anterior.

E. Duplicar el círculo las veces que sea necesario para generar los 10 pivotes.

F. Una vez definido el radio, coloreamos nuestro círculo con algún color diferente al de nuestro personaje para poder identificarlos mientras los ubicamos en el lugar que les corresponde y le otorgamos un nombre: pivot\_m0, m1, m2,..., m9 (Figura 7.6).

id	 Acept	ar
pivot_m0		

Figura 7.6. Valor del id de cada pivote.

G. La ubicación de los pivotes es dada en el siguiente orden:

Cuello: pivot\_m0 Hombro izquierdo: pivot m1 Codo izquierdo: pivot\_m2 Hombro derecho: pivot\_m3 Codo derecho: pivot\_m4 Cadera: pivot\_m5 Muslo izquierdo: pivot\_m6 Rodilla izquierda: pivot\_m7 Muslo derecho: pivot\_m8 Rodilla derecha: pivot\_m9

H. Los pivotes deberán quedar ubicados como en la Figura 7.7.

I. Guardamos el archivo como: nombre\_pivoteado.svg



Figura 7.7. Posición de los pivotes.

#### VIII. AGRUPACIÓN

Una vez que se han identificado cada una de las partes del esqueleto de la marioneta y se han asignado los pivotes; se crearán los grupos. Dichos grupos son etiquetas que permiten, como su nombre lo indica, agrupar los fragmentos de código de la imagen SVG según su pertenencia a las distintas secciones del cuerpo.

Para crear los grupos se siguen los siguientes pasos:

A. Utilizar el Editor XML para identificar cada una de las partes de la marioneta a través de la etiqueta <svg:path> y reubicarlas de acuerdo al grupo que corresponda.

B. Seleccionar todos los elementos que conformarán un grupo, para ello utilizar la tecla shift o del teclado.

C. Una vez que todos los path's son seleccionados ir al menú Objeto en la opción Agrupar (Figura 8.1), o bien, utilizar el comando ctrl + g.



Figura 8.1. Agrupación de todos los elementos de la cara.

D. Dar un valor al identificador del grupo de acuerdo a la Tabla 8.1. Aplicar los cambios dando clic en el botón Aceptar.

m0	pivot_m0	
	Contorno de la cara	
	Cara	
	Cabello	
	Cejas	
	Contorno de los ojos	
	Párpados	
	Iris	
	Nariz	
	Воса	
	Lengua	
	Cuello	
m1	pivot_m1	
	Contorno del hombro izquierdo	
	Hombro izquierdo	
	Brazo izquierdo	
	m2	pivot_m2
		Contorno del codo izquierdo
		Codo izquierdo
		Antebrazo izquierdo
		Contorno de la muñeca izquierda
		Muñeca izquierda
		Mano izquierda
m3	pivot_m3	
	Contorno del hombro derecho	
	Hombro derecho	
	Brazo derecho	
	m4	pivot_m4
		Contorno del codo derecho
		Codo derecho

Tabla 8 1	Elementos	ane	conforman	cada	aruno
1 abia 0.1.		que	comonnan	caua	grupo.

		Contorno de la muñeca derecha
		Muñeca derecha
		Mano derecha
m5	pivot_m5	
	Contorno de la cadera	
	Cadera	
m6	pivot_m6	
	Contorno de la cadera izquierda	
	Cadera izquierda	
	Muslo izquierdo	
	m7	pivot_m7
		Contorno de la rodilla izquierda
		Rodilla izquierda
		Pierna izquierda
		Contorno del tobillo izquierdo
		Tobillo izquierdo
		Pie izquierdo
m8	pivot_m8	
	Contorno de la cadera derecha	
	Cadera derecha	
	Muslo derecho	
	m9	pivot_m9
		Contorno de la rodilla derecha
		Rodilla derecha
		Pierna derecha
		Contorno del tobillo derecho
		Tobillo derecho
		Pie derecho

E. Hacer lo mismo con los demás elementos de la marioneta hasta tener los 10 grupos (Figura 8.2).

⊡Editor XML (Shift+Ctrl+X)	A A
<pre> svg:path id="Brazo_lzq" inkscape:label=""&gt; svg:path id="Brazo_lzq" inkscape:label=""&gt; </pre>	Nembro X Valor id m0
<pre><svg:path id="Hombro Izq" inkscape:label="">      </svg:path></pre> <pre></pre> <pre></pre> <pre></pre>	
<svg:path id="Cara_Contorno" inkscape:label=""> <svg:path id="Cuello" inkscape:label=""></svg:path></svg:path>	
<svg:path id="Cara" inkscape:label=""> <svg:path id="Boca" inkscape:label=""></svg:path></svg:path>	
<svg:path id="Nariz" inkscape:label=""> <svg:path id="Ojo_Der_Contorno" inkscape:label=""></svg:path></svg:path>	
<svg:path id="Ojo_Izq_Contorno" inkscape:label=""> <svg:path id="Ojo_Der_Iris" inkscape:label=""></svg:path></svg:path>	id Aceptar m0
<svg:path id="Ojo_Der_Parpado" inkscape:label=""> <svg:path id="Ojo_Izq_Iris" inkscape:label=""></svg:path></svg:path>	
Ha seleccionado el atributo id. Cuando termine, pulse Ctrl+Enter para gu	uardar los cambios.

Figura 8.2. Codificación XML para agrupación de elementos.

F. Crear el grupo "puppet" que incluya a los 10 grupos anteriores a él, m0 – m9, para tener todo en un solo conjunto y quede armada la marioneta.

G. Si el personaje no tiene movimiento en alguna articulación, se crea el grupo correspondiente a dicha articulación pero solamente se incluye el pivote de la misma y lo que incluía el grupo se incluye en el grupo puppet (ninguna parte de la marioneta debe quedar fuera de esté).

H. Esto se conoce como "Cancelación de Pivote".

**APÉNDICE C** 

MANUAL

PROCESO DE INSTALACIÓN DE UN SERVIDOR WEB EN RASPBERRY PI

# INSTITUTO TECNOLÓGICO DE CHIHUAHUA

### DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

## MANUAL

### PROCESO DE INSTALACIÓN DE UN SERVIDOR WEB EN RASPBERRY PI

LABORATORIO DE APRENDIZAJE MÓVIL

Autores:

Ing. Elisabet González Juárez

M.C. Alberto Pacheco González



Versión 1.4 Enero 2017

#### I. INSTALACIÓN DEL SISTEMA OPERATIVO RASPBIAN

La instalación del sistema operativo se puede hacer mediante dos maneras, utilizando directamente el instalador de Raspbian o utilizando NOOBS.

#### 1.1 Descarga de Raspbian

A. Ir al sitio web <u>www.raspberrypi.org/downloads/</u> y dar clic sobre el ícono de Raspbian como se ve en la Figura 1.1.



Figura 1.1. Selección de Raspbian en el sitio de descargas de Raspberry Pi.

B. Elegir la versión Raspbian Jessie con Pixel, ver Figura 1.2.

#### RASPBIAN

**Raspbian** is the Foundation's official supported operating system. You can install it with <u>NOOBS</u> or download the image below and follow our <u>installation guide</u>.

Raspbian comes pre-installed with plenty of software for education, programming and general use. It has Python, Scratch, Sonic Pi, Java, Mathematica and more.

The Raspbian with PIXEL image contained in the ZIP archive is over 4GB in size, which means that these archives use features which are not supported by older unzip tools on some platforms. If you find that the download appears to be corrupt or the file is not unzipping correctly, please try using <u>7Zip</u> (Windows) or <u>The</u> <u>Unarchiver</u> (Macintosh). Both are free of charge and have been tested to unzip the image correctly.



Figura 1.2. Sitio de descargas de Raspbian Jessie.

C. Seleccionar la opción Download ZIP, ver Figura 1.3.



Figura 1.3 Descarga del instalador de Raspbian Jessie.

D. Una vez descargado el instalador abrir la carpeta donde se haya guardado el archivo .zip y descomprimirlo. La carpeta deberá contener algo similar a lo de la Figura 1.4



Figura 1.4. Carpeta de archivos del instalador de Raspbian Jessie.

E. Formatear una tarjeta de memoria micro SD (*Secure Digital*) de al menos 4 GB de capacidad, como la de la Figura 1.5, al sistema de archivos FAT32, ver Figura 1.6



Figura 1.5. Tarjeta de memoria micro SD.

Capacidad:		
1.08 GB		•
Sistema de	archivos	
FAT32		•
amaño de	unidad de asignación	
4096 bytes	3	-
itiqueta del RECOVERN	l volumen	
Etiqueta del RECOVERI	l volumen (	
Etiqueta del RECOVER Opciones	l volumen ( de formato	
Etiqueta del RECOVER Opciones Ø Format	l volumen r de formato to rápido un disco de inicio de MS-DOS	
Etiqueta del RECOVER Opciones Ø Format	l volumen f de formato to rápido un disco de inicio de MS-DOS	
Etiqueta del RECOVER Opciones Ø Format	l volumen r de formato to rápido un disco de inicio de MS-DOS	
tiqueta del RECOVER Opciones Format Crear u	l volumen r de formato to rápido un disco de inicio de MS-DOS	

Figura 1.6. Cuadro de diálogo para dar formato a una tarjeta SD.

F. Para montar el instalador de Raspbian en la tarjeta micro SD utilizando Linux o Mac OS, seguir las instrucciones descritas en los siguientes enlaces:

Linux: <u>https://www.raspberrypi.org/documentation/installation/installing-images/linux.md</u>

Mac OS: <u>https://www.raspberrypi.org/documentation/installation/installing-images/mac.md</u>

Para Windows continuar a la sección 1.2

#### 1.2 Descarga de Win32 Disk Imager

A. Ir al sitio web <u>http://sourceforge.net/projects/win32diskimager/</u> y descargar el programa Win32 Disk Imager dando clic en el botón verde, ubicado en la parte superior derecha, ver Figura 1.7

iummary	Files	Reviews	Support	Wiki	Feature Requests	Bugs	Code	Mailing Li
★ 4.0 S ♦ 88,25 ∰ Last	Stars (80) 58 Downloa Update: 20	<mark>ds (</mark> This We 15-11-27	ek)		SF Download	oad the unamed s	sequel here	
V Twee	Gal	20	Me queta		an 1997		Brows	e All Files

Figura 1.7. Sitio de descarga de Win32 Disk Imager

B. Una vez descargado el instalador abrir la carpeta donde se haya guardado el archivo .exe y ejecutarlo. Aparecerá un cuadro de diálogo como el de la Figura 1.8. Dar clic sobre el botón Sí.



Figura 1.8. Cuadro de diálogo para dar permisos a Win32 Disk Imager.
C. A continuación se mostrará el programa de instalación. Dar clic sobre la opción Next, ver Figura 1.9.



Figura 1.9. Programa de instalación de Win32 Disk Imager.

D. Enseguida, verificar que la casilla l accept the agreement esté seleccionada, ver Figura 3.10. Dar clic en el botón Next.



Figura 1.10. Aceptación de los términos de licencia.

E. En la siguiente pantalla dar clic en el botón Next, ver Figura 3.11.

24	Setup - Win32DiskImager 🚽 🗖 🗙
	Select Destination Location Where should Win32DiskImager be installed?
	Setup will install Win32DiskImager into the following folder.
	To continue, dick Next. If you would like to select a different folder, dick Browse.
	C:\Program Files (x86)\ImageWriter Browse
	At least 44.9 MB of free disk space is required.
	< Back Next > Cancel

Figura 1.11. Selección de la ubicación destino para la instalación de Win32 Disk Imager.

F. En la siguiente pantalla dar clic en el botón Next, ver Figura 3.12.

1	Setup - Win32DiskImager 🚽 🗖 🗡
	Select Start Menu Folder Where should Setup place the program's shortcuts?
	Setup will create the program's shortcuts in the following Start Menu folder.
	Emage Writer Browse
_	
	< Back Next > Cancel



G. En la siguiente pantalla, verificar que la casilla Create a desktop icon esté seleccionada. Dar clic en el botón Next, ver Figura 1.13.



Figura 1.13. Creación de un ícono en el escritorio para Win32 Disk Imager.

H. A continuación, dar clic sobre el botón Install para comenzar con la instalación de Win32 Disk Imager, ver Figura 1.14.

24	Setup - Win32DiskImager -		×
	Ready to Install Setup is now ready to begin installing Win32DiskImager on your computer.		
	Click Install to continue with the installation, or click Back if you want to review or change any settings.		
	Destination location: C:\Program Files (x86)\ImageWriter Start Menu folder: Image Writer Additional tasks: Additional icons: Create a desktop icon	^	T.
	< >	×	
	< Back Install	Cano	:el

Figura 1.14 Confirmación de la instalación de Win32 Disk Imager.

I. Una vez que se haya completado la instalación de Win32 Disk Imager, verificar que la casilla Launch Win21DiskImager esté seleccionada, posteriormente dar clic sobre el botón Finish, ver Figura 1.15.



Figura 1.15. Finalización de la instalación de Win32 Disk Imager.

### 1.3 Montaje de la imagen de Raspbian en la memoria SD

A. Ejecutar el programa Win32 Disk Imager dando clic derecho sobre el acceso directo creado en el escritorio y seleccionando la opción Ejecutar como administrador, ver Figura 1.16.



Figura 1.16. Opción Ejecutar como administrador.

B. A continuación se abrirá el programa Win32 Disk Imager, tal y como se muestra en la Figura 1.17.

🔇 Win32 Disk Ima	ige,🍱		• • •	ā			_		X
Image File									Device
									[E:\] ▼
Copy MD5 Has	sh:								
Version: 0.9.5	Car	ncel	][	Read	)[	Wr	ite	][	Exit

Figura 1.17. Cuadro de diálogo del programa Win32 Disk Imager.

C. Una vez abierto el programa dar clic sobre el ícono de la carpeta que se muestra en la Figura 1.18.

🎨 Win32 Disk Ima	ige, 💷	120	•33	8		Ð	-		x
Image File									Device
									[E:\] ▼
Copy MD5 Has	sh:								
Version: 0.9.5	Car	ncel	)[	Read		W	rite	][	Exit
		_	_	_	_			_	

Figura 1.18. Ícono para seleccionar la imagen a copiar.

D. Ubicar el lugar donde se encuentra la imagen del sistema operativo, seleccionarlo y dar clic sobre el botón Abrir, ver Figura 1.19.

rganizar 🔻 Nueva carpeta		iii 🔹 🗖 🗎
<ul> <li>Mis vídeos</li> <li>node_modules</li> <li>OneDrive</li> <li>Tracing</li> <li>Vinculos</li> <li>VintualBox VMs</li> <li>Xilinx</li> <li>Equipo</li> <li>Windows 7 (C:)</li> <li>boot (E:)</li> <li>Documentos (I:)</li> <li>Programas (I:)</li> </ul>	Fecha de modifica Tipo 10/01/2017 05:59 Irfan	No hay ninguna vista previa disponible.
Raspberry Pi     2016-05-27-ra     2017-01-11-ra     NOOBS v1 9 -		

Figura 1.19 Ubicación del archivo de instalación de Raspbian Jessie.

E. Una vez que se seleccionó el archivo se mostrará en la pantalla del programa Win32 Disk Imager la ruta del archivo del instalador de Raspbian, ver Figura 1.20.

Image File							Devio
ry Pi/2017-01-11-ra	spbian-je	essie/20	17-01-11-	raspbia	n-jessie.	img 📔	[E:\]
Copy MD5 Has	h:						
Copy MD5 Has	h:						
Copy MD5 Has	h:						
Copy MD5 Has	h:						

Figura 1.20. Selección del archivo de la imagen de Raspbian Jessie.

F. Ya que se haya seleccionado la imagen del archivo seleccionar la unidad de memoria micro SD en donde se escribirán los archivos del instalador (Figura 1.21).

Image File					Device
ry Pi/2017-01-11-ra	spbian-jessie/	2017-01-11-	aspbian-jessi	e.img 📔	E:\]
- Internet in the second secon					
Copy MD5 Has	h:				
Copy MD5 Has	h:				
Copy   [] MD5 Has Progress	h:				
Copy   [] MD5 Has	h:				
Copy MD5 Has	h:	Dead	1	- 10	Evit

Figura 1.21 Selección de la unidad de almacenamiento.

G. Para verificar que se esté seleccionando la unidad correcta abrir la carpeta Equipo (Figura 1.22).



Figura 1.22. Carpeta Equipo.

H. El siguiente paso es escribir la imagen de instalación en la memoria micro SD, para ello, dar clic en el botón Write (Figura 1.23).

🍤 Win32 Disk Ima	ige, 💷 💷 🛛			
Image File				Device
rry Pi/2017-01-11-ra	spbian-jessie/20	017 <mark>-01</mark> -11-rasp	bian-jessie.img	[M:\] •
Copy MD5 Has Progress	sh:			
Version: 0.9.5	Cancel	Read	Write	Exit
Write data in 'Imag	ge File' to 'Devi	ce'	2012	e - /



I. En el cuadro de diálogo que aparece (Figura 1.24) dar clic sobre el botón Yes.



Figura 1.24. Confirmación de escritura en la memoria micro SD.

J. A continuación, comenzará la escritura de la imagen en la unidad de almacenamiento, el progreso se irá mostrando como se muestra en la Figura 1.25.

🧤 Win32 Disk Im	age, 🎟 📧					23
Image File					Dev	vice
erry Pi/2017-01-11-ra	aspbian-jessie/2	017-0 <mark>1-</mark> 11-	-raspbian-	jessie.img	E:\	•
Copy MD5 Ha	sh:				,	
			20102		1	14%
Version: 0.9.5	Cancel	Read		Write	Exit	
7.2324MB/s						

Figura 1.25. Progreso de escritura del instalador de Raspbian en la memoria micro SD.

K. Una vez que se haya completado la escritura, aparecerá un mensaje de confirmación, ver Figura 1.26.



Figura 1.26. Mensaje de confirmación de escritura terminada.

### 1.4 Instalación del sistema

A. Una vez hecho lo anterior, insertar la memoria micro SD en la tarjeta Raspberry Pi, ver Figura 1.27.



Figura 1.27. Ranura para tarjeta micro SD en la tarjeta Raspberry Pi.

B. Conectar el cable HDMI a la tarjeta Raspberry Pi y al monitor, también conectar el cable de red, ver Figura 1.28.



Figura 1.28. Conexión del cable HDMI al Raspberry Pi.

C. Así mismo conectar el ratón y el teclado en los puertos USB, ver Figura 1.29.



Figura 1.29. Puertos USB de la tarjeta Raspberry Pi.

D. Conectar el adaptador de corriente al Raspberry Pi y a la toma eléctrica, ver Figura 1.30.



Figura 1.30. Conexión del adaptador de corriente del Raspberry Pi.

E. Al conectar el adaptador de corriente eléctrica la tarjeta Raspberry Pi iniciará y en la pantalla se mostrará un cuadro de diálogo como el de la Figura 1.31, en donde se solicita un reinicio para configurar el sistema operativo.



Figura 1.31. Mensaje de reinicio del sistema de la tarjeta Raspberry Pi.

F. Después de que se haya reiniciado el sistema, comenzarán a cargarse los archivos de Raspbian, mientras eso sucede aparecerá un cuadro de diálogo con el logo de Raspberry Pi (Figura 1.32).



Figura 1.32. Pantalla de inicio para acceder a Raspbian.

G. Cuando el sistema operativo se haya terminado de configurar e iniciar, se mostrará el escritorio de Raspbian (Figura 1.33), hasta este paso ya se completó la instalación del sistema operativo.



Figura 1.33. Escritorio del sistema operativo Raspbian con Pixel.

# II. CONFIGURACIÓN DE CONEXIÓN SSH

Para poder trabajar con la tarjeta Raspberry Pi sin la necesidad de utilizar periféricos como ratón, teclado o pantalla se usa una conexión remota por medio del protocolo SSH. Dicha configuración se hace mediante los siguientes pasos.

## 2.1 Habilitación de conexión remota vía SSH

A. Abrir la terminal, dando clic en el ícono el que se encuentra en el menú superior. Al hacer esto aparecerá una pantalla como la de la Figura 2.1.

				pi@raspl	berrypi: ~			×
File	Edit	Tabs	Help					
pi@ra	aspber	гурі:-	- <b>S</b>					

Figura 2.1. Terminal de Raspbian.

B. Una vez abierta la sesión de la terminal escribir **sudo raspi-config** y presionar la tecla del teclado Enter, ver Figura 2.2.

File Edit Tabs Help pi@raspberrypi:~ \$ sudo raspi-config	
pi@raspberrypi:~ \$ sudo raspi-config	
	Í
	l
	l
	l
	l

Figura 2.2. Comando raspi-config.

C. A continuación aparecerá la pantalla de la Figura 2.3.



Figura 2.3. Panel de configuración de Raspbian.

D. Estando en el menú de configuración, desplazarse mediante las flechas del teclado

↑↓ hasta llegar a la opción Advanced Options, ver Figura 2.4.

pi@	)raspberrypi: ~		>
ile Edit Tabs Help			
Raspberry Pi Software Co	nfiguration Tool (raspi-config) 🛏		
1 Expand Filesystem 2 Change User Password 3 Boot Options 4 Internationalisation Options 5 Enable Camera 6 Overclock	Ensures that all of the SD card s Change password for the default u Configure options for start-up Set up language and regional sett Enable this Pi to work with the R Configure overclocking for your P		10 12 12 12 12 12 12 12 12 12 12 12 12 12
7 Advanced Options	Configure advanced settings		ġ
8 About raspi-config	Information about this configurat		
<select></select>	<finish></finish>		

Figura 2.4. Opción de configuración para acceder a las opciones avanzadas.

E. Presionar la tecla Enter del teclado para que aparezca el menú de la Figura 2.5.

Raspber	rry Pi Software	Configuration Tool (raspi-config)	_	
A1 Overscan		You may need to configure oversca		
A2 Hostname	20	Set the visible name for this Pi		
A3 Memory Spl	it	Change the amount of memory made	1988	
A4 SSH		Enable/Disable remote command lin	1262	
A5 VNC		Enable/Disable graphical remote a	8 188	
AD SPI		Enable/Disable automatic loading	2 198	
AR Serial		Enable/Disable shell and kernel m	82 :23	
A9 Audio		Force audio out through HDMI or 3	2 2	
AA 1-Wire		Enable/Disable one-wire interface	1	
	<select></select>	<back></back>		

Figura 2.5. Menú de opciones avanzadas.

F. En el menú de opciones avanzadas desplazarse mediante las flechas del teclado
 ↑↓ hasta llegar a la opción SSH, ver Figura 2.6.

	pi@raspberrypi: ~		
le Edit Tabs Help			
			-
Raspberry Pi Sc	oftware Configuration Tool (raspi-config) ⊨ 🗕		 -
A1 Overscan	You may need to configure oversca	1	
A2 Hostname	Set the visible name for this Pi	2 198	
A3 Memory Split	Change the amount of memory made	195	
A4#SSH	Enable/Disable remote command lin		
AS VNC	Enable/Disable graphical remote a	201 00	
A0 5P1	Enable/Disable automatic loading	2010	
A/ IZC	Enable/Disable automatic loading	20.00	
A0 Audio	Endole/Disable Shell and Kernel m	101	
AA 1 Wire	Epoble (Disoble one wire interface		
W 1-4116			
<5016	ert> <back></back>		
		_	
			Ī

Figura 2.6. Opción de configuración para habilitar SSH.

G. Presionar la tecla Enter del teclado, enseguida aparecerá el cuadro de diálogo de la Figura 2.7.

	pi@raspberrypi: ~		×
File Edit	Tabs Help		
			1
	Would you like the SSH server to be enabled?		
	<yes></yes>		
	L]		
			U

Figura 2.7. Mensaje de confirmación para la habilitación del SSH.

H. Seleccionar la opción Yes y presionar la tecla Enter del teclado para confirmar la habilitación del servidor SSH, ver Figura 2.8.

		pi@raspberrypi: ~		×
File	Edit	Tabs Help		
		Would you like the SSH server to be enabled?		
		<yes> <iio></iio></yes>		

Figura 2.8. Habilitación de SSH.

I. Una vez hecho lo anterior aparecerá un mensaje de confirmación, ver Figura 2.9.



Figura 2.9. Aviso de habilitación de SSH.

J. Presionar la tecla Enter del teclado para volver al menú principal (Figura 2.3).

K. Para finalizar, seleccionar la opción Finish (Figura 2.10) y presionar la tecla Enter del teclado.

pi@raspberrypi: ~						
File E	dit Tabs Help					
	Raspberry Pi Software Con	figuration Tool (raspi-config) 📙			7	
1 2 3 4 5 6 7	Expand Filesystem Change User Password Boot Options Internationalisation Options Enable Camera Overclock Advanced Options	Ensures that all of the SD card s Change password for the default u Configure options for start-up Set up language and regional sett Enable this Pi to work with the R Configure overclocking for your P Configure advanced settings			The second second	
8	About raspi-config	Information about this configurat				
	<select></select>	<pre>Keinish&gt;</pre>			State of the second sec	
-			_	_		

Figura 2.10. Opción para salir del menú de configuración.

### 2.2 Obtención de IP local

Para poder establecer la conexión vía SSH, es necesario conocer la IP de la tarjeta Raspberry Pi; esto se hace utilizando la terminal.

- A. Repetir el paso A de la sección 2.1.
- B. Escribir ifconfig en la terminal (Figura 2.11).



Figura 2.11. Comando para obtener la dirección IP.

C. Presionar la tecla Enter del teclado para que aparezcan los datos de red de la tarjeta Raspberry Pi y anotar la IP que aparezca en el campo **inet** addr de eth0, ver Figura 2.12.

	pi@raspberrypi:~ 🗕 🗖	×
File Ed	dit Tabs Help	
pi@rasp eth0	<pre>berrypi:- \$ ifconfig Link encap.Ethernet Hwaddr b8:27:eb:3b:2a:e4 inet addr:192.168.0.17 Bcast:192.168.0.255 Mask:255.255.255.0 ineto addr. feooosfd.614a:ee8d:4ba5/64 Scope:Link UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1 RX packets:10410 errors:0 dropped:0 overruns:0 frame:0 TX packets:7282 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000 RX bytes:13610453 (12.9 MiB) TX bytes:783765 (765.3 KiB)</pre>	
10	Link encap:Local Loopback inet addr:127.0.0.1 Mask:255.0.0.0 inet6 addr: ::1/128 Scope:Host UP LOOPBACK RUNNING MTU:65536 Metric:1 RX packets:200 errors:0 dropped:0 overruns:0 frame:0 TX packets:200 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1 RX bytes:16656 (16.2 KiB) TX bytes:16656 (16.2 KiB)	
pi@rasp	berrypi:- \$	

Figura 2.12. Datos de configuración de red de la tarjeta Raspberry Pi.

### 2.3 Descarga de PuTTy

A. Ir al sitio web <u>www.putty.org</u> (Figura 2.13) para descargar el programa PuTTy.

( ) ( www.putty.org	C Q Bush 🗘 🗘 🖨 🕈 🖉 🚍
	Download PuTTY PuTTY is an SSH and telnet client, developed originally by Simon Tatham for the Windows platform. PuTTY is open source software that is available with source code and is developed and supported by a group of volunteers. You can download PuTTY <u>here</u> .
2 <del></del>	Below suggestions are independent of the authors of PuTTY. They are not to be seen as endorsements by the PuTTY project.
	Bitvise SSH Client Bitvise SSH Client is an SSH and SFTP Client for Windows. It is developed and supported professionally by Bitvise. The SSH Client is robust, easy to install, easy to use, and supports all features supported by PuTTY, as well as the following: • single-click Remote Deachop tunneling; • auto-reconnecting capability; • dynamic port forwarding through an Integrated proxy; • an FTP-to-SFTP protocol bridge. Bitvise SSH Client is free to use. You can download it here.
	Bitvise SSH Server Bitvise SSH Server is an SSH, SFTP and SCP server for Windows. It is robust, easy to install, easy to use, and works well with a variety of SSH clients, including Bitvise SSH Client, OpenSSH, and PUTTY. The SSH Server is developed and supported professionally by Bitvise. You can <u>download Bitvise SSH Server here</u> .

Figura 2.13. Sitio web PuTTy.

B. Dar clic sobre el enlace de descarga, debajo de Download PuTTy, ver Figura 2.14.

Download PuTTY PuTTY is an SSH and telnet client, developed originally by Simon Tatham for the Windows by a group of volunteers. You can download PuTT here.
Below suggestions are independent of the authors of PuTTY. They are r.
<ul> <li>Bitvise SSH Client</li> <li>Bitvise SSH Client is an SSH and SFTP client for Windows. It is developed and supported p supported by PuTTY, as well as the following:</li> <li>graphical SFTP file transfer;</li> <li>single-click Remote Desktop tunneling;</li> <li>auto-reconnecting capability;</li> <li>dynamic port forwarding through an integrated proxy;</li> <li>an FTP-to-SFTP protocol bridge.</li> <li>Bitvise SSH Client is free to use. You can download it here.</li> </ul>
<b>Bitvise SSH Server</b> Bitvise SSH Server is an SSH, SFTP and SCP server for Windows. It is robust, easy to insta PuTTY. The SSH Server is developed and supported professionally by Bitvise. You can <u>download Bitvise SSH Server here</u> .

Figura 2.14. Enlace de descarga para PuTTy.

C. A continuación, se abrirá una nueva página web (Figura 2.15); estando en esa página, buscar el enlace de descarga para Windows<sup>®</sup>.

Binaries			
The latest releas	e version (beta 0.67)		
This will general	ly be a version we think is reasonably	likely to work well. If	you have a problem with the relea
For Windows on	Intel x86		
PuTTY:	putty.exe	(or by FTP)	(signature)
PuTTYtel:	puttytel.exe	(or by FTP)	(signature)
PSCP:	pscp.exe	(or by FTP)	(signature)
PSFTP:	psftp.exe	(or by FTP)	(signature)
Plink:	plink.exe	(or by FTP)	(signature)
Pageant:	pageant.exe	(or by FTP)	(signature)
PuTTYgen:	puttygen.exe	(or by FTP)	(signature)
A .ZIP file conta	ining all the binaries (except PuTT)	tel), and also the he	lp files
Zip file:	putty.zip	(or by FTP)	(signature)
A Windows MS	l installer package for everything ex	cept PuTTYtel	
Installer:	putty-0.67-installer.msi	(or by FTP)	(signature)
Legacy Inno Set	up installer. <u>Reportedly insecure</u> ! U	se with caution, if the	e MSI fails.
Legacy installer:	putty-0.67-installer.exe	(or by FTP)	(signature)

Figura 2.15. Sitio de descarga de PuTTy para Windows®.

D. Una vez que se haya descargado y ejecutado el archivo de instalación, se abrirá el programa (Figura 2.16).

- Seecion	Basic options for your P	UTTY enseion					
→ Session → Logging → Terminal → Keyboard → Bell → Features → Window → Appearance → Behaviour → Translation → Selection	Specify the destination you want to connect to Host Name (or IP address) Port Connection type: Raw Telnet Rlogin SSH Serial Load, save or delete a stored session Saved Sessions						
<ul> <li>Connection</li> <li>− Connection</li> <li>− Data</li> <li>− Proxy</li> <li>− Telnet</li> <li>− Rlogin</li> <li>⊕ SSH</li> <li>− Serial</li> </ul>	Close window on exit:	Load Save Delete					

Figura 2.16. Sitio de descarga de PuTTy para Windows®.

APÉNDICE D

## EVALUACIÓN DE LOS PROCEDIMIENTOS MD-01 Y MD-02



Figura D1. Cantidad de muestras de acuerdo al método utilizado.



Figura D2. Cantidad de muestras de acuerdo al grupo de edad.

Total	03:12:59	02:25:29	03:00:00	02:18:34	05:04:00	04:18:00	03:06:00	01:57:34	01:18:24	02:15:52	01:35:51	01:41:00	02:03:00	00:57:00	02:22:00	01:30:00	02:19:00	02:58:00	02:55:00	02:29:21
Correcciones	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00	00:15:00	00:10:00	00:00:00	00:15:00	00:04:00	00:14:00	00:30:00	00:00:00	00:14:40
Agrupación	00:00:00	00:00:00	01:15:00	01:30:46	00:42:00	00:20:00	00:00:00	00:48:28	00:20:02	01:00:00	00:19:35	00:20:00	00:24:00	00:00:00	00:30:00	00:22:00	00:10:00	00:15:00	00:00:00	00:35:29
Asignación de pivotes	01:59:02	01:30:00	01:15:00	00:21:43	00:28:00	00:28:00	00:00:00	00:20:59	00:16:44	00:16:52	00:15:00	00:20:00	00:19:00	00:00:00	00:12:00	00:02:00	00:22:00	00:25:00	00:00:00	00:33:31
Sectorizado de la imagen	00:00:00	00:00:00	00:00:00	00:00:00	01:43:00	01:40:00	01:13:00	00:10:54	00:11:19	00:40:00	00:14:51	00:30:00	00:18:00	00:00:00	00:30:00	00:16:00	00:00:00	00:15:00	00:30:00	00:37:51
Coloreado de la imagen	00:53:59	00:45:00	00:15:00	00:15:06	00:56:00	00:53:00	00:16:00	00:16:31	00:10:48	00:04:00	00:09:28	00:02:00	00:07:00	00:00:00	00:15:00	00:10:00	00:39:00	00:20:00	00:10:00	00:22:16
Vectorización de la imagen	00:19:58	00:10:29	00:15:00	00:10:59	00:44:00	00:44:00	01:00:00	00:14:36	00:19:31	00:15:00	00:11:23	00:60:00	00:18:00	00:00:00	00:20:00	00:18:00	00:21:00	00:13:00	00:25:00	00:21:36
Digitalización de la imagen	00:00:00	00:00:00	00:00:00	00:00:00	00:04:00	00:03:00	00:13:00	00:04:36	00:00:00	00:00:00	00:02:10	00:01:00	00:05:00	00:30:00	00:00:00	00:00:00	00:00:00	00:00:00	01:30:00	00:16:58
Trazado de la imagen	00:00:00	00:00:00	00:00:00	00:00:00	00:27:00	00:10:00	00:24:00	00:01:30	00:00:00	00:00:00	00:23:24	00:01:00	00:22:00	00:27:00	00:20:00	00:13:00	00:33:00	01:00:00	00:20:00	00:21:41
No. de muestra	1	2	з	4	5	9	7	8	6	10	11	12	13	14	15	16	17	18	19	Total

Tabla D1. Tiempos de diseño para cada una de las etapas del proceso de elaboración de una marioneta digital.