



**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y  
COMPUTACIÓN**

**TESIS DE GRADO**

**”DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA PARA LA  
MANIPULACIÓN DE OBJETOS VIRTUALES POR MEDIO DE UN  
LENGUAJE DACTILOLÓGICO”**

**Previa a la Obtención del Título de Ingeniero en Computación  
Especialización Sistemas Multimedia**

**PRESENTADA POR:  
JOSÉ ANTONIO ORAMAS MOGROVEJO**

**GUAYAQUIL – ECUADOR**

**2008**

## **AGRADECIMIENTO**

*A mi familia*

*por ser un apoyo incondicional.*

*A mis profesores*

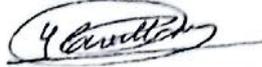
*por guiarme durante mi formación académica*

*A mis valiosos amigos.*

*A Dios.*

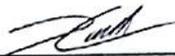
## DEDICATORIA

*Con cariño y aprecio,  
a todas las personas  
que de una u otra forma  
creyeron en mi.*

**TRIBUNAL DE GRADO****PRESIDENTE**

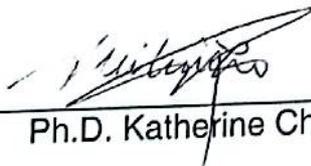
---

Ing. Holger Cevallos Ulloa

**DIRECTOR DE TESIS**

---

Ph.D. Xavier Ochoa

**MIEMBROS PRINCIPALES**

---

Ph.D. Katherine Chiluiza



---

Msc. Guido Caicedo

## DECLARACIÓN EXPRESA

"La responsabilidad por los hechos, ideas y doctrinas expuestas en esta tesis, nos corresponden exclusivamente; y, el patrimonio intelectual de la misma, a la Escuela Superior Politécnica del Litoral"

(Reglamento de exámenes y títulos profesionales de la ESPOL)



José Antonio Oramas Mogrovejo

## **RESUMEN**

En este documento se presentará la propuesta de un sistema para la manipulación de objetos virtuales por medio de un lenguaje dactilológico. La tesis está separada en 5 capítulos donde se tratarán diversos aspectos del sistema como las bases sobre la cual parte la propuesta realizada, su diseño, desarrollo y los resultados obtenidos durante las diferentes pruebas realizadas antes y después de su implementación.

En el Capítulo 1 se presentan los antecedentes necesarios para entender el paradigma de interacción base del sistema a proponer. Se presentan los diferentes paradigmas de interacción empleados actualmente. Además se describen algunas alternativas comerciales y estudios en torno a dispositivos con interfases hápticas y se analiza la necesidad de plantear un diseño que permita la implementación de bajo costo de este tipo de dispositivos.

En el Capítulo 2 se presentan los requerimientos y características que el sistema a proponer debe satisfacer así como el alcance y las limitaciones que la solución propuesta posee. Adicionalmente se muestran las alternativas disponibles para la implementación del sistema propuesto como un análisis de las mismas.

En el Capítulo 3 se cubre el diseño del sistema, se presentan los componentes y módulos que el sistema propuesto posee, tanto en su parte de software como de hardware, así la forma de interacción de estos. En este capítulo se hace énfasis en las características que cada uno de estos componentes y módulos deben poseer para poder cumplir con los requerimientos planteados en el capítulo anterior.

En el Capítulo 4 se describe la etapa de implementación del sistema. En este capítulo se muestran los dispositivos de hardware empleados para la implantación de cada uno de los componentes del sistema así como el sistema operativo, entornos de trabajo, librerías y demás herramientas de software empleadas para la implementación de los módulos del sistema. Finalmente se muestra el algoritmo encargado del manejo del lenguaje dactilológico como el de la aplicación de manipulación de imágenes.

En el Capítulo 5 se presenta el conjunto de pruebas realizadas durante y después del proceso de implementación, junto con los resultados obtenidos y el análisis respectivo de los mismos.

Para finalizar, se obtienen las conclusiones con respecto a los resultados obtenidos luego del proceso de implementación y prueba del sistema propuesto. Se analizan sus fortalezas y debilidades, y se proponen mejoras.

## **ÍNDICE GENERAL**

AGRADECIMIENTO .....	ii
DEDICATORIA .....	iii
TRIBUNAL DE GRADO .....	iv
DECLARACIÓN EXPRESA .....	v
RESUMEN .....	vi
ÍNDICE GENERAL.....	viii
INDICE DE FIGURAS.....	ix
INDICE DE TABLAS .....	x
INTRODUCCIÓN.....	1
ANTECEDENTES Y JUSTIFICACIÓN .....	2
<i>Interacción Hombre Ordenador</i> .....	2
Definición, Objetivos e Historia .....	2
Paradigmas de Interacción .....	7
<i>Evolución de Tecnologías Hápticas</i> .....	10
<i>Trabajo e Investigación relacionada</i> .....	13
Soluciones Comerciales .....	17
<i>Definición del problema</i> .....	22
<i>Conclusiones</i> .....	24
ANÁLISIS.....	26
Requerimientos Funcionales .....	26
Requerimientos No Funcionales.....	27
<i>Características y Alcance de la solución propuesta</i> .....	28
Componentes del Sistema.....	32
<i>Limitaciones del Sistema</i> .....	33
<i>Pruebas y Comportamientos Esperados</i> .....	35
<i>Alternativas para desarrollo del sistema</i> .....	37
Hacks de Wiimote .....	37
TouchLib .....	39
Bespoke 3DUI XNA .....	40
HandVu.....	40
Análisis de Alternativas.....	42
<i>Conclusiones</i> .....	46
DISEÑO .....	47
<i>Diseño de Hardware</i> .....	47
<i>Módulos del Sistema</i> .....	51
<i>Arquitectura del Sistema</i> .....	62
<i>Diseño de Software</i> .....	64
<i>Modelo de Datos</i> .....	64
<i>Conclusiones</i> .....	66

IMPLEMENTACIÓN.....	67
<i>Tecnología.....</i>	67
Sistema Operativo .....	67
Lenguaje de Programación.....	68
<i>Implementación de los Componentes del sistema .....</i>	69
Componentes de Hardware.....	69
Componentes de Software .....	73
<i>Algoritmo de Detección, Seguimiento y Reconocimiento de posturas     dactilológicas de HandVu.....</i>	77
<i>Procesamiento de la Aplicación de manipulación de imágenes.....</i>	78
<i>Conclusión.....</i>	83
PRUEBAS Y RESULTADOS .....	85
<i>Prueba 1 (P-1).....</i>	86
<i>Prueba 2 (P-2).....</i>	89
<i>Prueba 3 (P-3).....</i>	91
<i>Prueba 4 (P-4).....</i>	94
<i>Prueba 5 (P-5).....</i>	96
<i>Prueba 6 (P-6).....</i>	99
<i>Conclusiones.....</i>	101
CONCLUSIONES .....	101
REFERENCIAS .....	109
ANEXOS.....	114

## **INDICE DE FIGURAS**

<b>Gráfico 1.1.</b> MERL DiamondTouch .....	18
<b>Gráfico 1.2.</b> MS Surface.....	19
<b>Gráfico 1.3.</b> LM3LABS UbiqWindow .....	19
<b>Gráfico 1.4.</b> iWindow .....	20
<b>Gráfico 1.5.</b> Muro Sensible de Perceptive Pixel.....	21
<b>Gráfico 1.6.</b> React-In InviDesk .....	21
<b>Gráfico 1.7.</b> Tactiva TactaPad.....	22
<b>Gráfico 2.1.</b> Diagrama de Casos de Uso del Sistema propuesto. ....	27
<b>Gráfico 2.2.</b> Componentes del sistema.....	32
<b>Gráfico 2.3.</b> Seguimiento de dedos de la mano usando Wiimote.....	38
<b>Gráfico 2.4.</b> Seguimiento de dedos de la mano usando TouchLib.....	39
<b>Gráfico 2.5.</b> Seguimiento de dedos de la mano usando HandVu .....	41
<b>Gráfico 3.1</b> Flujo de Funcionamiento del MAI. ....	52
<b>Gráfico 3.2.</b> Flujo de funcionamiento del MDS.....	53
<b>Gráfico 3.3.</b> Flujo de Funcionamiento del MRG. ....	54
<b>Gráfico 3.4.</b> Flujo de Funcionamiento del MA. ....	55
<b>Gráfico 3.5.</b> Módulos del Sistema propuesto. ....	56
<b>Gráfico 3.6.</b> Interacción entre los Módulos del Sistema. ....	58
<b>Gráfico 3.8.</b> Diagrama de Secuencia de la correcta ejecución de una operación.....	59
<b>Gráfico 3.8.</b> Diagrama de Secuencia del inicio de interacción con el sistema. ....	60
<b>Gráfico 3.9.</b> Diagrama de Secuencia del proceso en el cual la postura dactilológica realizada por el usuario no es reconocida.....	61
<b>Gráfico 3.10.</b> Diagrama de Secuencia del proceso en el cual el gesto dactilológico realizado por el usuario no es reconocido.....	62
<b>Gráfico 4.1.</b> HandVu .....	75
<b>Gráfico 4.2.</b> Aplicación de Manipulación de Imágenes. ....	76
<b>Gráfico 4.3.</b> Flujo de Procesamiento inicial de la aplicación durante el renderizado de imágenes. ....	80
<b>Gráfico 4.4.</b> Flujo de Procesamiento para la manipulación de imágenes en la aplicación. ....	83
<b>Gráfico 5.1.</b> Implantación del sistema empleando un monitor LCD. ....	86
<b>Gráfico 5.2.</b> usuarios de Prueba .....	87
<b>Gráfico 5.3.</b> Tiempo de detección de las manos de los usuarios. ....	88
<b>Gráfico 5.4.</b> Posturas que por defecto son reconocidas por HandVu. ....	89
<b>Gráfico 5.5.</b> Tiempo de detección de las posturas seleccionadas. ....	90
<b>Gráfico 5.6.</b> Aplicación de Pruebas VisualPad.....	93
<b>Gráfico 5.7.</b> Grafico de resultados de pruebas de VisualPad .....	93
<b>Gráfico 5.8.</b> Tiempo de detección sobre fondos dinámicos. ....	95
<b>Gráfico 5.9.</b> Tiempo de detección de las manos del usuario sobre fondos dinámicos, empleando una cámara infrarroja. ....	97
<b>Gráfico 5.10.</b> Comparación entre los tiempos de detección de las cámaras empleadas. ....	98
<b>Gráfico 5.11.</b> Comparación entre los tiempos de detección de las cámaras empleadas. ....	99
<b>Gráfico 5.12</b> Tiempo que le tomó a cada usuario escalar y trasladar una imagen.....	100

## **ÍNDICE DE TABLAS**

<b>Tabla 2.1.</b> Tabla de Comparación de las herramientas existentes. ....	42
<b>Tabla 4.1.</b> Tabla de cámaras usadas para la implementación del dispositivo de captura .....	70
<b>Tabla 4.2.</b> Dispositivos empleados para la implementación del Área de Trabajo.....	72
<b>Tabla 4.3.</b> Especificaciones del computador empleado.....	73
<b>Tabla 4.4.</b> Herramientas empleadas para la implementación de los módulos del sistema. ....	76
<b>Tabla A.1:</b> Tiempo de detección de las manos del usuario empleando HandVu.....	115
<b>Tabla B.1:</b> Tiempo de reconociendo de las posturas detectadas por HandVu en una prueba realizada a diez usuarios. ....	115
<b>Tabla C.1:</b> Tiempo que le tomó los usuarios realizar la tarea de desplazamiento de imágenes empleando el prototipo de pruebas VisualPad.....	116
<b>Tabla D.1:</b> Comparación de tiempo de detección de las manos de dos usuarios, empleando una cámara de luz visible (CLV) y una cámara de luz infrarroja (CLI) sobre un televisor CRT.....	116
<b>Tabla E.1:</b> Tiempo que le tomó a dos usuarios realizar la tarea de modificación de tamaño y traslación de imágenes usando el sistema propuesto.....	117

## **INTRODUCCIÓN**

En sus primeros años, los computadores se caracterizaron no sólo por ser dispositivos que poseían un gran tamaño sino también por la complejidad que tenía su uso. En estos tiempos, las computadoras era controladas por operadores especializados por medio cables y switches, pasando por tarjetas perforadas y poco a poco interfaces más intuitivas y fáciles de usa, que surgieron con la popularidad del computador personal.

Con el objetivo de mejorar la interacción de los usuarios con sus computadores, se plantearon diferentes paradigmas y estilos de interacción. Uno de estos paradigmas es el de interfaces hápticas, popularizado recientemente en productos que emplean superficies sensitivas que gozan de alta intuitividad y facilidad de uso, pero poseen la desventaja de poseer un costo elevado.

El presente trabajo muestra el desarrollo de una superficie sensitiva que responde a gestos y posturas dactilológicas realizada por el usuario. Adicionalmente, se propone un diseño para el desarrollo de estas superficies, que apunta a la disminución de costos en base a la reutilización de componentes de hardware y software. Se muestran resultados obtenidos luego de la evaluación de un prototipo de pruebas. Finalmente se presentan conclusiones de este trabajo y recomendaciones para futuras mejoras del diseño propuesto.

# **CAPÍTULO I.**

## **ANTECEDENTES Y**

### **JUSTIFICACIÓN**

#### **Interacción Hombre Ordenador**

#### **Definición, Objetivos e Historia**

Interacción Persona Ordenador, Interacción Hombre Computador o HCI, como es conocida mundialmente por sus siglas de su traducción en inglés, es la disciplina que se enfoca en el estudio de la interacción entre las personas y los sistemas computacionales. Su objetivo principal es mejorar esta interacción haciendo que los sistemas computacionales sean más usables, de manera que aumente la productividad de las personas al trabajar con ellos. La Asociación de Maquinaria Computarizada (ACM) (4) define HCI como “La disciplina encargada del diseño,

evaluación e implementación de sistemas computacionales interactivos para uso humano y del estudio de lo que los rodea". La ACM hace énfasis en HCI como una ciencia que analiza ambos aspectos: humano y computador, en conjunto. Esta es una de las razones primordiales por lo cual HCI es estudiada con un enfoque distinto dependiendo de la ciencia. Desde el contexto humano, HCI es complementada por ciencias como: comportamiento humano, psicología, ciencias cognitivas, teoría de comunicación, disciplinas de diseño gráfico e industrial, lingüísticas, ciencias sociales y rendimiento humano. Ciencias en el contexto de computadores y maquinaria comprende: gráficos por computador, sistemas operativos, lenguajes de programación, y desarrollo de ambientes. Ciencias que complementan a HCI de una manera relevante.

Como es planteado por el Centro para el Estudio de Librerías Digitales (CSDL) de la Universidad A&M de Texas en su curso de HCI (CPSC 436) (6), el modelo conceptual de HCI contempla cuatro elementos: (a) las personas; usuarios del sistema, (b) la tarea; diferentes pasos a realizar para llevar a cabo una o mas actividades, (c) el ambiente; aspectos físicos, organizacionales y sociales del ambiente y, (d) la tecnología: cualquier artefacto con el cual se interactúa.

A diferencia de los aspectos del ambiente, y de la tarea; la interacción entre personas y tecnología se realiza por medio de un componente un tanto implícito: la Interfaz. Esta interfase está

formada de varios componentes, entre estos se puede nombrar, Interfaces de hardware: teclado, mouse, touchpads, lápices, etc. e interfaces de software como la Interfaz Gráfica de Usuario (GUI).

Dando una mirada al pasado, podemos encontrar dispositivos que incluyeron interfaces de hardware diferentes a la combinación tradicional de teclado y mouse que han sido empleados por años. Un buen ejemplo de uno de estos dispositivos puede ser encontrado en Sketchpad (36). Sketchpad fue el sistema desarrollado por Ivan Sutherland en 1963 como parte de su investigación para la obtención de su Ph. D. en el Instituto de Tecnología de Massachusetts (MIT). Este sistema fue uno de los primeros en emplear una GUI, utilizaba una matriz para trazado y la, recientemente inventada, pluma de luz. Por medio de esta pluma el usuario podía trazar gráficas y manipular los trazos existentes, convirtiéndose en uno de los primeros programas de trazado asistido por computador (CAD). Adicionalmente, gracias a la orientación a objetos con la que estaba programado, permitía la modificación masiva de objetos previamente instanciados. Este sistema propuesto por Sutherland no solo lo hizo merecedor al premio Turing en 1988 sino que también fue la prueba para demostrar que los gráficos por computador pueden ser empleados tanto para propósitos artísticos como técnicos. El trabajo de Sutherland fue seguido durante los años 60 por una serie de investigaciones que perseguían la combinación del hombre y los sistemas computacionales. Entre estos sistemas se puede

encontrar a: Man-Computer Symbiosis (21) y Augmentation of Human Intellect (8). Man-Computer Symbiosis fue propuesto por Licklider en 1960. Planteaba una cooperación interactiva en la cual el hombre se encargaría de formular hipótesis, determinar el criterio y realizar evaluaciones. Por otro lado, los computadores se encargarían de realizar el trabajo rutinario que debe ser realizado para preparar el camino para análisis profundos y pensamiento científico. Augmentation of Human Intellect fue propuesto en 1962 por Engelbart; en esta investigación se planteaba un marco de trabajo para poder desarrollar sistemas que se encarguen de mejorar la efectividad de manejo de conocimiento por los seres humanos. Mas tarde en los años 70, la metáfora del escritorio apareció gracias a la propuesta de Tim Morr y Larry Tesler mientras trabajaban en el Xerox PARC. Esta metáfora es la que rige la mayoría de las interfaces gráficas de usuario de los sistemas operativos de computadores personales; Consiste en tratar al monitor de computador como si fuera la parte superior del escritorio del usuario, sobre el cual documentos y carpetas pueden ser colocados. Un documento puede ser abierto en una ventana, que representa la copia de papel del documento colocado sobre el escritorio. Adicionalmente, están disponibles pequeñas aplicaciones llamadas accesorios del escritorio. Esta metáfora ha sido empleada por años desde su aparición en 1973 en la Xerox Alto. En los últimos dos años, gracias a los avances en el campo de gráficos por computador, una versión mejorada y

mas inmersiva llamada BumpTop (1) fue propuesta por Anand Agarawala y Ravin Balakrishnan en el 2006. BumpTop es un prototipo que enriquece la metáfora del escritorio con efectos de física y reconocimiento de gestos. La inclusión de efectos de física dio un movimiento y flujo mas realista a los objetos y herramientas sobre el escritorio. También mejoraron las acciones que los usuarios pueden realizar, usando gestos como medio de ingreso de acciones y comandos que los usuarios pueden darle a las aplicaciones. Sin embargo, algo que se debe tener presente es que aún cuando se mejore la GUI mediante el uso de una metáfora apropiada o mediante la implementación de varios efectos para hacerla mas inmersiva, si su contraparte de hardware no permite al usuario hacer las acciones que naturalmente hace para realizar la actividad requerida en la vida real, el sistema permanece como una actividad ligada a un computador. Para evitar caer en este problema, Agarawala y Balakrishnan decidieron aplicar una interfase de pluma en el lado del hardware. Para extender este acercamiento de interacción utilizando una pluma, ellos sintetizaron técnicas de cruzado de metas con arrastrar, soltar y selección de lazo. BumpTop es uno de los últimos pero no el único o último proyecto siguiendo esta línea de sistemas de interfaces naturales.

## Paradigmas de Interacción

Con el objeto de plantear métodos para el desarrollo de interfaces de hardware lo más natural posible, se plantearon los paradigmas de interacción. Con el pasar del tiempo y con el avance de la tecnología han aparecido cada vez mas paradigmas de interacción, ciertas formas de interacción que antes se consideraban paradigmas por separado se han unificado bajo un solo paradigma. Entre los paradigmas de Interacción mas reconocidos podemos encontrar los siguientes:

**Hipertexto, hipermedios e hiperenlaces:** este paradigma consiste en el uso de objetos, sean estos hipertextos; texto y/o , hipermedios; imágenes, videos, sonidos, etc. como enlaces hacia otros objetos relacionados. De esta manera se pueden ingresar los contenidos en una estructura relacional con el objetivo de lograr una navegación mas fluida y dinámica.

**Realidad Mediada por Computador:** este paradigma consiste en el uso del computador para modificar la percepción de la realidad. Este se divide en dos campos: Realidad Virtual (VR), que consiste en la creación de realidades alternas o ambientes simulados en donde se puede interactuar con objetos virtuales, y Realidad Aumentada (AR) cuyo objetivo es aumentar o enriquecer nuestra realidad con datos u objetos generados por el computador

**Audio como medio de Interacción:** este paradigma toma como base el uso de sonidos como medio para dar o recibir

instrucciones hacia y desde los sistemas computacionales. Está soportado por diferentes disciplinas: reconocimiento de diálogos, síntesis de diálogos, procesamiento de lenguaje natural y procesamiento de sonidos.

**Trabajo Colaborativo respaldado por Computador:** este paradigma realiza un análisis de todos los aspectos que rodean al trabajo colaborativo. Las herramientas disponibles dentro de este paradigma se ven divididas dependiendo de los factores que posibilitan el trabajo colaborativo: disponibilidad de las personas tanto en un lugar como en un momento dado.

**Computación Ubícua:** el paradigma de computación ubicua propone la integración de sistemas computacionales en objetos de uso diario. Sistemas que se encuentren tan inmersos en dichos objetos que las personas no noten que los están usando al realizar sus actividades diarias con dichos objetos.

**Computación Vestible:** el paradigma de computación vestible toma como base el uso de computadores que se usan como prenda de vestir, con el objetivo de dar soporte a las personas mientras estas tienen sus sentidos ocupados con el ambiente físico.

**Interfaz Cerebro Computador:** este paradigma plantea la implementación de una vía de comunicación directa entre la actividad cerebral o nerviosa de un ser humano, animal o una red neuronal viva. Los sistemas que se rigen por este tipo de paradigma se ven divididos por la dirección que sigue el flujo de

información. De esta forma, existen sistemas de una vía como de dos vías.

**Háptica y Teleháptica:** En el campo de la tecnología, Háptica se refiere al sistema que tiene como medio de interacción el tacto; esto incluye la aplicación de fuerzas, vibraciones y/o movimientos por parte del usuario. Por otro lado, teleháptica se refiere a la generación de impulsos sensibles al tacto por medio de un computador. Por lo general, este tipo impulsos generados por computador se utilizan como un medio de retroalimentación de un sistema computacional hacia el usuario. Un ejemplo común, dentro de la industria de los juegos de video, puede ser la tecnología Dual Shock de Sony empleada en el Play Station. En esta plataforma la retroalimentación es causada por medio de pequeños motores colocados dentro del control de juego. Mientras el jugador interactúa con un juego, el juego está programado para que dependiendo de ciertos eventos: colisiones, explosiones, etc. el motor gire haciendo vibrar el control en las manos del jugador. Haciendo de esta forma que la experiencia de juego sea más inmersiva. En comparación a la háptica, la teleháptica no es tan aplicada. Con los recientes avances de la tecnología, se puede decir que es un campo que está en crecimiento.

## **Evolución de Tecnologías Hápticas**

Las tecnologías que usan interfaces hápticas han sido empleadas desde varios años atrás. Una de las primeras aplicaciones de dispositivos hápticos fue la usada en grandes aviones modernos que utilizan servo sistemas para operar el sistema de control. Inicialmente estos sistemas de control eran de una vía en el sentido que las fuerzas aerodinámicas aplicadas a estos no podían ser percibidas en los controles, con la fuerza normal faltante simulada con pesos y dispositivos elásticos. En los primeros aviones livianos, que no poseían un servo sistema, a medida que el avión se aproximaba a una condición de pérdida de sustentación, se sentía el bofeteo aerodinámico en los controles del piloto, como advertencia de la peligrosa condición de vuelo. Esta vibración de controles no es percibida cuando se emplean sistemas de servo control. Para reemplazar esta advertencia, se emplea una masa giratoria no-balanceada para simular los efectos de un control simple. Esto es conocido como retroalimentación háptica y es muy similar a la tecnología DualShock de Sony, antes mencionada. Otro tipo de dispositivos que se basan en interfaces hápticas son las superficies sensitivas. Estas superficies son empleadas en diferentes disciplinas: diseño, arte, visualización de datos, etc. . Muchas veces son manipuladas por medio de herramientas adicionales como lápices, o elementos mas cercanos a la aplicación, como es el caso de metaDesk.

MetaDESK (13) es un sistema presentado por Hiroshi Ishii en 1997. Este sistema utiliza herramientas como lupas, cubos, visores, etc. para proveer una forma de interacción mas realista; utilizando elementos cuya forma de uso y funcionalidad que brindan es muy similar a los elementos existentes en el mundo físico.

TransBoard, ambientROOM y el ya mencionado metaDESK, son proyectos que fueron presentados en 1997. Ishii y Ulmer (13) utilizaron estos proyectos como la base con la que propusieron un nuevo paradigma de interacción cuya interfase constaba en medios tangibles, interfaces que luego fueron conocidas como interfaces tangibles de usuario (TUIs). Ellos enfatizaron la debilidad de HCI que se basa en GUIs que muestran la información como “bits pintados” en pantallas rectangulares; restringiéndose a un conjunto limitado de canales de comunicación. “Las GUIs quedan cortas al intentar cubrir la capacidad de los sentidos humanos y las habilidades que las personas han desarrollado durante una vida de interacción con el mundo físico”, este es el problema que Ishii y Ulmer apuntaron a resolver y fue la base de su investigación. Adicionalmente su investigación se enfoca en TUIs que “aumentarán” los objetos del mundo físico con información digital, con el objetivo de usar el mundo en sí como una interfase, mezclando el mundo real con el ciberespacio.

Actualmente, como resultado de varias investigaciones en el campo de las TUIs y con el fácil acceso que tienen las personas a la tecnología; varios dispositivos que emplean TUIs están disponibles en el mercado. Dispositivos orientados a usuarios regulares como iPhone de Apple (2) , GPSs, tablas digitalizadoras empleadas por artistas, etc. y otros dispositivos ofertados como innovadoras soluciones empresariales como los muros interactivos de Perceptive Pixel (32), Surface de Microsoft, (27) InviDesk de React-In (34), entre otros, orientados para exhibiciones o demostraciones. Sin embargo, estas últimas opciones, las que son ofertadas a empresas y corporaciones, son bastante costosas si las comparamos con las alternativas orientadas a usuarios regulares. Magic Wall, como fue llamado el muro interactivo empleado por CNN para la transmisión de las primarias estadounidenses, fue desarrollado por Perceptive Pixel y tuvo un costo aproximado de 100,000 dólares. Por otro lado, la primera versión comercial de Surface tuvo un costo de aproximadamente 10,000 dólares. Esto representa un problema para usuarios regulares como para pequeñas empresas quienes no pueden costear uno de estos dispositivos y quienes tienen las mismas necesidades y obtendrían las mismas ventajas de poder emplear este tipo de dispositivos.

## Trabajo e Investigación relacionada

En el campo de superficies sensitivas se han ejecutado diferentes proyectos y realizado diferentes investigaciones. Algo que debemos puntualizar, son los enfoques que se ha dado al medio que se utiliza para la interacción. Entre las propuestas planteadas existen sistemas en los cuales se interactúa por medio de herramientas externas: lápices, guantes, lupas, cubos, tarjetas, etc. , como es el caso de metaDESK , mientras que en otros sistemas no se emplea herramienta alguna; sistemas en los cuales la interacción se realiza empleando las manos. Según lo resaltado por Manresa [2000] (24), “las interfaces de usuario sin el uso de guantes o herramientas adicionales y fondos controlados son mas naturales. La existencia de estos dos factores en un sistema producen, como consecuencia, una limitación tanto al usuario como a la interfaz”. Por este motivo, el sistema a desarrollar como producto de esta tesis, se enfocará a permitir interacción netamente por medio de las manos.

En 1996 Quek y Zhao (33), y mas tarde en el 2003 Wu y Balakrishnan (44) emplearon una técnica de aprendizaje inductivo con el objetivo de reducir el tiempo de procesamiento computacional requerido para el reconocimiento. El problema detectado en su propuesta era el amplio set que requería su algoritmo para entrenamiento.

Siguiendo otro enfoque, Nolker y Ritter [1996] (30) utilizaron Modelos Ocultos de Markov para la identificación de gestos simples. Aún cuando el enfoque inicial era diferente, el requerimiento de un amplio set para entrenamiento persistía.

Mas tarde, una nueva perspectiva fue propuesta por Lien y Huang [1998] (22) empleando algoritmos que se basan en modelos. Esta perspectiva aumentó el nivel de precisión en la detección, sin embargo resultó en un procesamiento no era apto para reconocimiento en tiempo real.

Adicionalmente han habido líneas de investigación que se enfocaron en el seguimiento y reconocimiento de gestos de la mano como medio para el control de aplicaciones. Durante el 2007, Wang, Zhang y Dai realizaron experimentos (41) haciendo la detección y seguimiento de la “deformable mano humana”, como ellos lo plantearon, para usarla como medio de interacción continua en base a gestos dactilológicos. Sus experimentos se concentraron en la detección y seguimiento de las puntas de los dedos para, dependiendo de la posición de estos, reconocer gestos. Esta investigación fue similar a la realizada en el 2000 por Manresa, quien propuso el uso de gestos dactilológicos como medio de interacción con un juego de video. Otra alternativa fue propuesta por Mathias Kolsch [2004] en la cual proponía un algoritmo (17) que tomaba como base características KLM y las mantenía por medio de un algoritmo para manejar comportamientos de rebaño. Este algoritmo fue propuesto para

emplearlo en sistemas de computación vestible para detectar y seguir objetos articulados, en este caso las manos del usuario. La diferencia en base a lo previamente expuesto es que esta investigación se concentraba solo en el uso de gestos y posturas dactilológicas para la interacción.

En lo relacionado con interfaces interactivas en base a algoritmos de visión por computador, ha habido una gran variedad de trabajos. VIDEOPLACE (18), uno de los primeros proyectos de investigación, era un ambiente virtual de pruebas implementado por Kruger en los 70s. Su idea era la creación de una realidad artificial que rodearía a los usuarios y que respondería a sus movimientos sin la necesidad de dispositivos adicionales como visores o guantes especiales. Dentro de VIDEOPLACE los usuarios eran capaces de interactuar con el sistema con su cuerpo, una funcionalidad remarcable era la posibilidad de usar las manos para seleccionar objetos y los dedos para dibujar en las pantallas. Otro proyecto relevante fue el que resultó en DigitalDesk (42). DigitalDesk era un escritorio regular con funcionalidades aumentadas que no solo proveía interacción con las manos, sino con documentos impresos también. Una cámara y un proyector se encontraban colocados en la parte superior del escritorio, apuntando perpendicularmente a la superficie de este. Mientras el proyector se encargaba de mostrar objetos virtuales con los que el usuario podía interactuar, la cámara se encargaba de detectar tanto las manos del usuario como documentos

impresos. Una versión similar, pero mas portable, de DigitalDesk fue Tele-Graffiti (29). La diferencia entre estos sistemas es que Tele-Graffiti apunta mas a ser una superficie con una interacción cuya base son los trazados. Luego un sistema con funcionalidad similar, pero empleando las manos como medio de interacción, apareció: FingerMouse (28). FingerMouse es una alternativa al mouse como dispositivo apuntador. El objetivo de este proyecto era desarrollar una interfase para apuntar que disminuya el tiempo tomado por el usuario al mover su mano del teclado al mouse. Este sistema utilizaba algoritmos de visión por computador para detectar y seguir las manos del usuario sobre el teclado desde una cámara colocada en la parte superior. Este sistema reconocía un gesto dactilológico como apuntador, y mapeaba la posición de la mano con la del apuntador en la pantalla y permitía dar clicks con el uso del botón shift. Aún cuando FingerMouse alcanzó su objetivo, posee una debilidad, está limitado a una sola acción: un solo click. Visual Touchpad (23) es un dispositivo de entrada que permite interacción con el uso de las dos manos. Usa una técnica de visión estereoscópica para detectar y seguir la posición y orientación de la punta de los dedos. Además provee información de contacto en el momento en que el usuario hace contacto con la superficie sensitiva. Para proveer retroalimentación al usuario, la GUI es aumentada con gráficas de las manos del usuario, capturadas por las cámaras, pero modificando su canal alfa para mostrarlas semitransparentes. La desventaja de este sistema es

su naturaleza de touchpad. Esto evita que el usuario pueda trabajar directamente sobre la información. En el 2007 el boom de los hacks de Wiimote inició con la propuesta de Johnny Lee (20). Lee propuso emplear el control remoto del Wii de Nintendo (Wiimote) como una cámara infrarroja de alto rendimiento para mejorar la detección. Los tres demos con los cuales Johnny Lee realizó su propuesta fueron: la creación de pizarras digitales, seguimiento de vista humana y detección de dedos usando una cinta reflectiva en la punta de estos. Aún cuando esta idea es uno de los hacks mas impresionantes de los últimos años, introduce un elemento no natural: el uso de cinta reflectiva.

## **Soluciones Comerciales**

Existen varios productos comerciales disponibles en el mercado que emplean un paradigma de interacción similar al expuesto previamente y al requerido para la solución a proponer. Sin embargo, debido al enfoque comercial que estos tienen, las especificaciones del funcionamiento interno de estos son confidenciales. Por este motivo, esta sección se limitará a dar una descripción superficial de los mismos.

**Diamond Touch:** Este sistema fue desarrollado en el 2001 en el Laboratorio de Investigación Eléctrica de Mitsubishi (MERL). Es una mesa sensitiva al tacto creada con el objetivo de permitir el trabajo colaborativo entre varios usuarios mientras mantienen

contacto visual con la información. Como es especificado en la página oficial del proyecto (25),



**Gráfico 1.1.** MERL DiamondTouch

DiamondTouch emplea proyección frontal para mostrar la información, adicionalmente posee antenas embebidas en la superficie sensitiva. Lo interesante de esta tecnología es que puede detectar múltiples acciones de un mismo usuario. Esto lo hace ya que en el momento en el usuario toca la superficie, una pequeña señal es enviada por medio del cuerpo del usuario a un receptor colocado en la silla de este.

**Surface:** es un sistema propuesto por Microsoft (27). El primer prototipo fue mostrado en el 2003 mientras que fue lanzado al mercado en el 2005. Este sistema tiene la forma de una mesa, para mostrar las aplicaciones emplea retroproyección por medio de un proyector que se encuentra alojado en la parte inferior de esta. Adicionalmente en la parte inferior, cinco cámaras infrarrojas detectan actividad en la superficie. Esta actividad puede darse

cuando un usuario toca la superficie, como cuando algún objeto, con alguna etiqueta predeterminada, es colocado sobre la mesa.



**Gráfico 1.2.** MS Surface

**UbiqWindow:** Este sistema mostrado por LM3LABS, es una pantalla con la que se interactúa utilizando gestos de la mano, algo que se debe tener presente es que no necesariamente debe existir contacto entre el usuario y la superficie de la pantalla para poder llevar a cabo la interacción. Esta característica le brinda a UbiqWindow (39) la posibilidad de ser colocado por detrás de vidrios o materiales transparentes para protegerlo. Actualmente es usado para presentaciones en centros comerciales y a diferencia de los sistemas antes mencionados, no tiene forma de mesa sino que se lo emplea generalmente en forma vertical apuntando hacia afuera en las ventanas de centros comerciales.



**Gráfico 1.3.** LM3LABS UbiqWindow

**iWindow:** El concepto detrás de iWindow (14) es muy similar al propuesto por UbiqWindow, una pantalla sensitiva colocada a disposición de las personas con la cual se puede interactuar a través de una ventana. Aunque los detalles de esta tecnología no están disponibles, se puede mencionar que para la detección de gestos del usuario empleando una lámina sensitiva. Esta lámina es colocada sobre un LCD o una pantalla de retroproyección para convertir dicho medio de salida en un medio sensitivo.

Adicionalmente iWindow provee la funcionalidad de correr aplicaciones existentes en diferentes formatos como Flash, PPT, etc.



**Gráfico 1.4.** iWindow

**Magic Wall:** es la forma en que fue llamada la pantalla táctil empleada por CNN durante la cobertura de las elecciones para la presidencia de Estados Unidos durante el 2008. Este sistema fue desarrollado por PerceptivePixel (32) empleando la tecnología de reflexión total interna (FTIR). Esta tecnología emplea un arreglo de leds colocados entre dos láminas. Este es un sistema de retroproyección, adicionalmente junto al proyector, se encuentra un grupo de cámaras que detectan la luz reflejada por los dedos

del usuario en el momento en que este entra en contacto con la superficie sensitiva.



**Gráfico 1.5.** Muro Sensible de Perceptive Pixel

**InviDesk:** El escritorio virtual interactivo o InviDesk (34) como es generalmente abreviado por su traducción al inglés, es un producto de la empresa alemana ReactIn. Su hardware tiene la forma de una mesa, funciona con proyección frontal, en este caso desde la parte superior de esta, y permite la interacción por parte de varios usuarios por medio de lápices inalámbricos. Esta última característica es lo que diferencia a InviDesk de otras alternativas; presenta una forma de interacción que se basa en lápices. Finalmente, al igual que las alternativas previamente presentadas, tiene una orientación hacia la presentación de datos y diseño.



**Gráfico 1.6.** React-In InviDesk

**TactaPad:** Este sistema (37) en el cual el usuario interactúa usando sus manos, se diferencia de los anteriores ya sigue el esquema de funcionamiento de un touchpad. Al igual que un touchpad, posee una superficie sensitiva al tacto, pero adicionalmente emplea técnicas de visión por computador para la detección de las manos de este. Para dar retroalimentación al usuario, TactaPad aumenta sobre la imagen mostrada en el monitor las manos del usuario con un leve nivel de transparencia; algo muy similar a lo visto en Visual TouchPad.



**Gráfico 1.7.** Tactiva TactaPad

### **Definición del problema**

Tomando en cuenta lo antes expuesto, se puede observar los beneficios de naturalidad y facilidad de uso que sistemas con interfaces hápticas poseen; sin embargo, se pueden detectar dos problemas que surgen en el momento de desarrollar una superficie sensitiva. Estos problemas dependen del enfoque que se tome durante la implementación del sistema. El primer enfoque se concentra en el hardware; son sistemas que utilizan un dispositivo de hardware especializado para poder detectar el lugar

de contacto del usuario con la superficie sensitiva. Este hardware especializado puede ser tanto una superficie sensible al tacto como una cámara especializada. El uso de este tipo de hardware provee una alta precisión para la detección del contacto del usuario, sin embargo el problema que se presenta al seguir este enfoque es el alto costo que este hardware especializado posee. El segundo enfoque cubre los sistemas que usan como base algoritmos de Visión por Computador. En este tipo de sistemas la interacción se realiza sobre cualquier superficie y es una cámara, por lo general una cámara web, la que sensa los movimientos del usuario; muy similar a lo visto en Visual TouchPad. Al emplear una cámara web los costos de hardware se reducen significativamente, lo que representa una de las ventajas de este enfoque, lo que cambia es que para el desarrollo se requiere un nivel de conocimiento avanzado de algoritmos de visión por computador. Esto implica un requerimiento cognitivo a los programadores e investigadores que no sean expertos en dicho campo de las ciencias en computación.

Considerando los puntos antes tratados, el objetivo de esta tesis es plantear un diseño que solucione los problemas de alto costo y complejidad de implementación de interfaces hápticas. Para esto se presenta la implementación de un sistema de relativo bajo costo cuyo paradigma base de interacción sea el de interfaces hápticas. Entre la variedad de estilos que se encuentran dentro de este paradigma, la solución propuesta seguirá el estilo de

interacción sobre superficies sensitivas sobre la cual el usuario ingresará instrucciones por medio de un lenguaje dactilológico. Para alcanzar este objetivo general, esta tesis seguirá los siguientes objetivos específicos:

- Análisis del estado de arte de sistemas con una forma de interacción similar a la propuesta.
- Análisis del diseño de la solución propuesta.
- Implementación de un sistema de Bits Tangibles a un costo relativamente bajo.
- Determinar la efectividad del reconocimiento del lenguaje dactilológico propuesto usando los algoritmos empleados en el sistema propuesto y los factores que influyen en la efectividad del mismo.
- Experimentar este tipo de interacción con diferentes perfiles de usuarios y determinar la aceptación de este tipo de interacción de parte de los mismos.
- Sentar las bases para futuros experimentos basados en formas de interacción alternativas.

## **Conclusiones**

En este primer capítulo se presentó una breve descripción de la evolución de las interfaces de usuario y de los diferentes paradigmas de interacción existentes en la actualidad. Se enfatizó en los hitos que ha tenido la evolución de interfaces hápticas, como en investigaciones realizadas y en soluciones comerciales

existentes en lo referente al estilo de interacción de superficies sensitivas. Luego, se especificó los problemas de los métodos empleados para la implementación de este tipo de superficies. Finalmente, este capítulo concluye presentando los objetivos que esta tesis persigue.

# CAPÍTULO II.

## ANALISIS

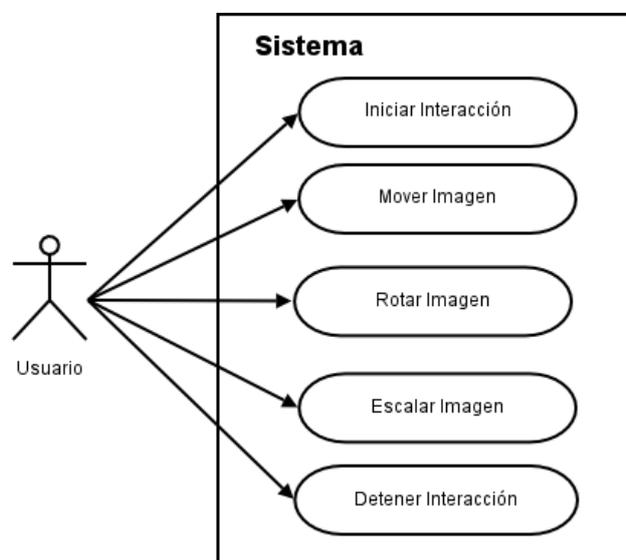
Ya que el producto a desarrollar en esta tesis está enfocado a ser un prototipo de pruebas y no una solución, no cuenta con un cliente en particular y necesidades del mismo. Sin embargo, existe un conjunto de requerimientos funcionales; tareas y características concretas que deben ser satisfechas por el sistema, y requerimientos no funcionales; características generales que debe poseer el sistema.

### **Requerimientos Funcionales**

- El sistema debe poder ser controlado por medio de posturas y gestos dactilológicos.
- La aplicación a emplear en el sistema permitirá la manipulación de objetos virtuales.

- La aplicación de pruebas debe permitir la traslación, rotación y cambio de tamaño de las imágenes mostradas en la interfaz de usuario.
- La aplicación de pruebas debe ser relativamente fácil de aprender a usar.

Los requerimientos de la aplicación de pruebas que va a correr en el sistema se muestran en el siguiente diagrama de casos de uso.



**Gráfico 2.1.** Diagrama de Casos de Uso del Sistema propuesto.

## Requerimientos No Funcionales

En base a los objetivos planteados en esta tesis y en base a ciertas características que sistemas computacionales similares requieren para su correcto funcionamiento, se requiere:

Desarrollar un sistema cuyo rendimiento lo haga apto para realizar procesamiento, funcionar satisfactoriamente y proveer retroalimentación en tiempo real.

Implementar un sistema cuya arquitectura permita la portabilidad del mismo a diferentes sistemas operativos y ambientes de programación.

Plantear un esquema de implementación de bajo costo para el sistema a implementar.

### **Características y Alcance de la solución propuesta.**

Al apuntar al desarrollo de una superficie sensitiva de relativo bajo costo, se debe especificar lo que este documento va a considerar como costo. Para la implementación del sistema propuesto, se va a considerar como costo: (a) los requerimientos económicos, (b) los requerimientos cognitivos. Requerimientos económicos se refiere al valor monetario necesario para adquirir equipos adicionales para la implementación del sistema. Equipos adicionales son los equipos con los cuales un usuario no cuenta, son equipos que se adquieren expresamente para la implementación del sistema. Requerimientos cognitivos se refiere a los conocimientos que se necesitan para la implementación del mismo. En lo referente a la consideración *a*, la estructura física del sistema se orientará a poseer un diseño que promueva la reutilización de elementos existentes para ser usados como componentes del sistema. En cuanto al factor *b*, el diseño del sistema se orientará a la utilización de librerías y marcos de trabajo existentes de manera que se facilite el desarrollo de aplicaciones para el sistema.

Para interactuar con el sistema el usuario utilizará sus manos y el ingreso de instrucciones se realizará mediante un lenguaje dactilológico. Un lenguaje o sistema dactilológico, es una forma de comunicación utilizando las manos; este comprende tanto posturas como gestos de la mano. Para evitar ambigüedades, cuando este documento utilice el término “Postura dactilológica”; se referirá a la posición y orientación de los dedos con respecto a la palma de la mano en un momento determinado. Cuando se utilice el término “Gesto dactilológico”; el documento se referirá a las siluetas o movimientos que realice el usuario con su mano como un todo. A diferencia de las posturas, los gestos dactilológicos son independientes del movimiento y/o posición de los dedos del usuario. La importancia del uso de este tipo de medio para la interacción del sistema se resume en los siguientes puntos:

***Tarea ajena a la realidad:*** Como fue enfatizado por Manresa en el 2000: “las interfaces sin el uso de guantes o elementos adicionales agregados al usuario y fondos controlados son mas naturales” (24). En este punto Manresa puntualiza en como ciertos sistemas obligan al usuario a usar implementos que, naturalmente, no utilizarían para realizar una tarea similar en el mundo físico, sin el uso del computador. Además agrega: “el uso de estos elementos limita tanto al usuario como a la interfaz”. Para sustentar esta afirmación en primer lugar se puede mencionar la interferencia que dispositivos mecánicos suelen

tener frente a campos electromagnéticos. En segundo lugar ciertos dispositivos limitan los grados de libertad que posee el usuario, como fue demostrado por Sturman y Zeltzer [1989] (35), existen dispositivos que generan interferencia en las acciones que desea realizar el usuario, a lo que ellos llamaron “ruido ergonómico”. En su estudio del uso de gestos y posturas dactilológicas como medio de interacción con un ambiente virtual, emplearon guantes DataGlove con el objetivo de sensar las posturas que realizaba el usuario con sus manos. Para poder medir dichas posturas empleaban las medidas de flexión de los dedos proveídas por dichos guantes. Durante sus pruebas detectaron como dichos guantes limitaban a los usuarios en cuanto a las posturas que deseaban realizar y además como los campos electromagnéticos afectaban las medidas sensadas por los guantes.

**Reutilización de conocimiento/habilidades:** Como fue puntualizado por Ishii [1997] (13) en la propuesta del paradigma de medios tangibles, “Las GUIs quedan cortas al intentar cubrir la capacidad de los sentidos humanos y las habilidades que las personas han desarrollado durante una vida de interacción con el mundo físico”. Al emplear las manos como “herramienta” de interacción se busca explotar las habilidades y conocimientos que han adquirido las personas sobre la forma en que son empleadas y referentes al significado que gestos y posturas de estas poseen. Durante su estudio Sturman y Zeltzer (1989) (35) demostraron

como la combinación de posturas y gestos dactilológicos forman parte de un conjunto de movimientos complejos que no son aprovechados por las metáforas empleadas tradicionalmente y que son expresiones familiares de la interacción social diaria.

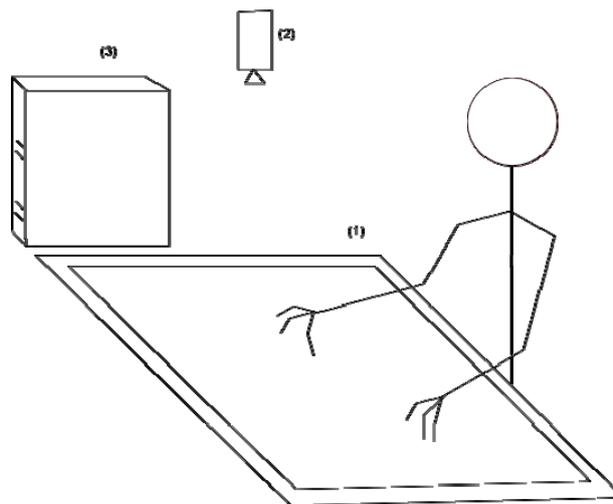
Como se mencionó anteriormente, la aplicación de prueba que va a correr en el sistema es una aplicación básica de manipulación de imágenes. Esta aplicación va a ser muy similar a las que se plantean en las alternativas comerciales. Entre las operaciones que podrá realizar el usuario sobre las imágenes se encuentran traslación, rotación y modificación de tamaño de las imágenes.

Al iniciar, la aplicación cargará un número determinado de imágenes. Cuando el usuario coloque la mano dentro el área de trabajo esta va a ser detectada y seguida. El sistema constantemente escaneará la mano del usuario para detectar tanto posturas como gestos dactilológicos. Las posturas van ser reconocidas por el sistema como una forma de indicar la instrucción a realizar, mientras que los gestos van a servir para medir el nivel en que se desee realizar dicha instrucción. Por esto, para la manipulación de imágenes, el usuario deberá colocar su mano sobre una imagen y realizar una postura determinada. Si la postura realizada es la de traslación, dependiendo del gesto realizado por el usuario la imagen se desplazará a medida que desplace su mano sobre el área de trabajo. Lo mismo ocurrirá si el usuario realiza la postura para indicar que desea escalar o rotar la imagen. Cuando el usuario desee "soltar" una imagen, deberá

realizar una postura para indicar dicha instrucción. En lo relacionado a la traslación de imágenes, se puede decir que el sistema realiza algo parecido a un drag & drop, pero en lugar de usar un click para seleccionar y soltar, utiliza una postura dactilológica. Cabe mencionar que dichas posturas y gestos son preseleccionados.

### Componentes del Sistema

El gráfico 2.2 muestra los componentes que conforman el sistema propuesto, entre estos componentes se encuentran: (1) área de trabajo, (2) dispositivo de captura, (3) computador y (4) software.



**Gráfico 2.2.** Componentes del sistema

El área de trabajo es la región donde se muestra la aplicación. Es aquí donde el usuario ingresa instrucciones por medio de un lenguaje dactilológico. Este lenguaje está compuesto tanto de posturas como de gestos que este realice con sus manos. Es por medio del área de trabajo que el usuario ingresa instrucciones y recibe retroalimentación visual de las acciones que realiza.

El dispositivo de captura se encarga de sensar las acciones que realiza el usuario sobre el área de trabajo. Es una cámara que se encuentra apuntando perpendicularmente al área de trabajo desde la parte superior de esta.

El computador está conectado tanto al dispositivo de captura como al hardware usado para implementar el área de trabajo. Se encarga de procesar las señales recibidas por el dispositivo de captura y enviar señales de retroalimentación al área de trabajo.

El componente de software que corre en el computador, por el tipo de procesamiento que realiza, se puede dividir en dos partes: la parte de detección y seguimiento; que se encarga de la detección e identificación del lenguaje dactilológico, y la parte de aplicación; encargada de manejar la lógica de la aplicación y presentar la retroalimentación adecuada en el área de trabajo.

### **Limitaciones del Sistema**

Debido a que el sistema propuesto es un prototipo de prueba tendrá las siguientes limitaciones:

***Aplicación Inicial Única:*** En su presentación inicial el sistema va a alojar una aplicación de software única. Como fue expuesto en la sección anterior, esta aplicación es de manipulación de imágenes; va a cargar inicialmente un número determinado de imágenes y va a permitir al usuario desplazarlas, rotarlas y modificar su tamaño. Todo trabajo realizado sobre el conjunto de imágenes es temporal; cada vez que el usuario reinicie la

aplicación, el sistema cargará nuevamente las imágenes en ubicaciones aleatorias con sus tamaños iniciales. El usuario no tendrá control sobre el número de imágenes a cargar ni se manejará persistencia de datos.

**Lenguaje Dactilológico Predeterminado:** El sistema a desarrollar va a reconocer un número preestablecido de posturas y gestos dactilológicos. Dichas posturas van a ser seleccionadas en base al conjunto de marcos de trabajo, librerías y demás herramientas de software empleadas para el desarrollo del sistema, considerando los resultados obtenidos durante las pruebas de dichos elementos. El lenguaje dactilológico a emplear en el sistema no va a necesariamente relacionarse con lenguajes dactilológicos o lenguajes de señas ya existentes; la implementación del sistema no persigue dicha relación.

**Reutilización de Hardware de Mejor Esfuerzo:** aún cuando el sistema busca la minimización de costos en los dos contextos antes mencionados, en el contexto de hardware, no se garantiza la adaptabilidad total a cualquier hardware existente. Durante el proceso de pruebas se identificará las características que el hardware que se desee reutilizar debe poseer. Esto se debe a que ciertos dispositivos no proveen o un correcto nivel de procesamiento o las condiciones necesarias para un aceptable funcionamiento del sistema.

## **Pruebas y Comportamientos Esperados**

Durante el proceso de desarrollo se van a realizar un conjunto de pruebas para verificar el funcionamiento de cada uno de los módulos del sistema. Adicionalmente, se plantea una prueba para verificar la usabilidad del sistema.

### **Prueba 1 (P-1)**

**Objetivo:** Verificar la efectividad de la librería/marco de trabajo seleccionado para realizar la detección y seguimiento de las manos del usuario.

**Comportamiento esperado:**

El programa de pruebas de la librería/marco de trabajo seleccionado detecta la mano del usuario.

Una vez que la mano del usuario ha sido detectada, el programa realiza el seguimiento de la misma.

### **Prueba 2 (P-2)**

**Objetivo:** Identificar las posturas dactilológicas que la librería/marco de trabajo seleccionado reconozca con mayor grado de precisión.

**Comportamiento esperado:**

El programa de pruebas de la librería/marco de trabajo seleccionado detecta las posturas dactilológicas predeterminadas.

El programa genera una retroalimentación para verificar que la postura dactilológica ha sido reconocida.

### **Prueba 3 (P-3)**

**Objetivo:** Evidenciar el correcto funcionamiento de la librería/marco de trabajo seleccionado al ser integrado a una aplicación de pruebas a ser desarrollada.

**Comportamiento esperado:**

El programa de pruebas de HandVu detecta las seis posturas dactilológicas predeterminadas.

El programa genera una retroalimentación para verificar que la postura dactilológica ha sido reconocida.

### **Prueba 4 (P-4)**

**Objetivo:** Evidenciar la precisión de detección de posturas dactilológicas en fondos no controlados.

**Comportamiento esperado:**

El programa de pruebas de la librería/marco de trabajo seleccionado detecta la mano del usuario y hace el seguimiento de las mismas.

Se espera un retardo en el tiempo de detección de las manos del usuario y en el tiempo de reconocimiento de posturas dactilológicas.

Dependiendo del resultado de esta prueba, modificaciones en la implementación del sistema y futuras pruebas sobre fondos dinámicos pueden realizarse.

### **Prueba 5 (P-5)**

**Objetivo:** Verificar la usabilidad del sistema al probar la aplicación de pruebas de manipulación de imágenes desarrollada.

**Comportamiento esperado:**

El sistema detecta las manos del usuario y hace el seguimiento de estas.

Al detectar las posturas dactilológicas empleadas en la aplicación de imágenes, el usuario podrá trasladar, rotar o cambiar el tamaño de la imagen sobre la cual realice dicha postura.

### **Alternativas para desarrollo del sistema**

Para el desarrollo de superficies sensitivas, nos vamos a enfocar en cuatro alternativas propuestas:

#### **Hacks de Wiimote**

En el año 2006 Nintendo lanzó al mercado su mas reciente consola de juegos, la Nintendo Wii. Esta consola incluía un gran cambio en lo que tiene que ver con jugabilidad. La jugabilidad usando la Nintendo Wii, adicionalmente de proveer una control inalámbrico como es tradicional en las consolas de juego de video, proveía la capacidad de interactuar con el juego realizando movimientos físicos. Para lograr esto, la consola incluye un dispositivo que no es otra cosa mas que un arreglo de leds. Estos leds sirven de punto de referencia para que el control remoto (Wiimote) pueda detectar su posición y enviar dichos datos vía bluetooth a la consola. Durante su investigación Johnny Lee

conectó un Wiimote a su computador e identificó características de la cámara infrarroja de alto rendimiento que este posee. Entre las características importantes constan:

- Resolución de 1024x768.
- Detección, desde hardware, de cuatro puntos de mayor Intensidad.
- Posee un acelerómetro que permite medir la aceleración en los tres ejes.

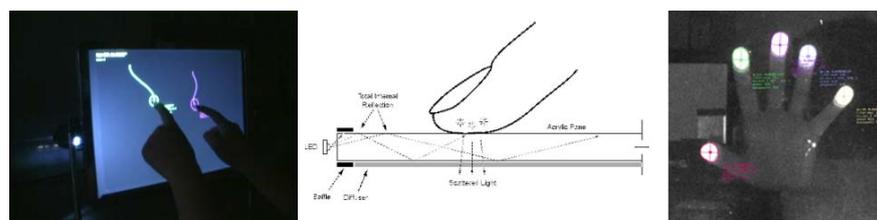
En base a esto, Lee propuso el uso del Wiimote para mejorar la interacción con sistemas computacionales. Entre sus propuestas se encuentran: la creación de pizarras digitales de bajo costo, visores para realidad virtual y seguimiento de dedos (gráfico 2.3). Desde la publicación de los videos de estas propuestas en YouTube en Diciembre del 2007, cada día son mas los proyectos que incluyen esta tecnología. Aunque en el momento de la propuesta Lee publicó el código que empleó bajo el sistema operativo MS Windows, existen códigos para el control del Wiimote desde otros sistemas operativos como MacOX y GNU/Linux.



**Gráfico 2.3.** Seguimiento de dedos de la mano usando Wiimote

## TouchLib

TouchLib (38) es una librería creada con el objetivo de facilitar el desarrollo de aplicaciones que requieran interacción múltiple de usuarios. Maneja la detección y seguimiento de manchas de luz infrarroja empleando la tecnología Reflexión Interna Total Frustrada (FTIR). En su plataforma de hardware requiere que el dispositivo de adquisición de imágenes y el proyector se encuentren apuntando en la dirección en la que se encuentra el usuario, específicamente en la parte posterior en la que el hace contacto con la superficie sensitiva. Esto hace que TouchLib se vea restringida a funcionar en sistemas de proyección posterior o inferior, como se puede ver en el gráfico 2.4. Para ser empleado en alguna aplicación, envía eventos tales como: dedo-presionando, dedo-levantado, dedo-trasladado, etc. por medio del protocolo TUIO, lo que le permite funcionar de forma distribuida. Durante su publicación solo se encontraba disponible para ambientes MS Windows; sin embargo, migraciones y pruebas en otros sistemas operativos están siendo realizadas en la actualidad.



**Gráfico 2.4.** Seguimiento de dedos de la mano usando TouchLib

## **Bespoke 3DUI XNA**

Luego de la innovadora propuesta de Lee, el marco de trabajo Bespoke 3DUI XNA (5) fue desarrollado por el laboratorio IS/UE.

Entre sus características principales podemos encontrar:

- Renderizado estereoscópico.
- Seguimiento óptico de cabeza con seis grados de libertad.
- Soporte para movimientos en tres dimensiones empleando el Wiimote.
- Un extensible reconocedor de gestos en tres dimensiones.

Este marco de trabajo provee un conjunto de clases y rutinas para el control del Wiimote desde un ambiente MS Windows. Esta es una librería OpenSource disponible bajo la Licencia Pública de Microsoft (MPL) (26). Está implementado sobre el Microsoft XNA Game Studio y soporta el renderizado de gráficos tanto en dos como en tres dimensiones.

Adicionalmente, se agregó funcionalidad para poder emplear tecnología FTIR sin embargo, como fue presentado previamente tanto el uso del Wiimote como el de la tecnología FTIR introduce limitaciones y elementos no naturales al sistema.

## **HandVu**

Mathias Kölsch y Matthew Turken desarrollaron HandVu (11) con el objetivo de ser usada como interfaz para el control de aplicaciones de computación vestible utilizando las manos como

medio directo de interacción. Esta librería tiene como base algoritmos de visión por computador para la detección y seguimiento de las manos del usuario y reconocimiento de posturas dactilológicas. Está diseñado para adquirir imágenes de una cámara apuntando al espacio adelante del usuario desde una vista superior, como se puede observar en el gráfico 2.5. Esta cámara debe proveer una resolución mínima de 320x240. Esta librería está escrita en C++ y emplea funciones de la librería OpenCV para el procesamiento de imágenes. Entre la funcionalidad prestada provee el reconocimiento de seis diferentes tipos de posturas de la mano y el uso de ambas manos. Provee funcionamiento distribuido por medio de un servidor de gestos que envía eventos en formato ASCII via TCP/IP. Adicionalmente provee un demo de integración con ARToolkit (3). ARToolkit permite el fácil desarrollo de aplicaciones de realidad aumentada. La integración de HandVu con este toolkit promueve el uso de las manos para interactuar con los objetos virtuales aumentados a la realidad.



**Gráfico 2.5.** Seguimiento de dedos de la mano usando HandVu

## Análisis de Alternativas

Entre las cuatro alternativas presentadas existen factores que deben ser considerados. Para su selección se realizó un análisis que consideraba los siguientes criterios: (a) Naturalidad de Uso, (b) Facilidad de Implementación, (c) Rendimiento, (d) Portabilidad, e (e) Independencia de Hardware.

En la tabla 2.1 se muestra un resumen del análisis realizado. Cada alternativa fue calificada en la escala del 1 al 3, mientras mayor es la calificación obtenida por una de las alternativas, mayor es la funcionalidad o ventaja obtenida dentro del criterio analizado. Tomando en cuenta los totales obtenidos, HandVu es la opción que brinda mayor ventaja para la implementación del sistema a proponer en este documento.

	Naturalidad de Uso	Facilidad de Implementación	Rendimiento	Portabilidad	Independencia de Hardware	total
Hacks de Wiimote	1	3	3	2	1	10
Touchlib	3	3	3	2	2	13
Bespoke 3DUI XNA	3	3	3	1	1	11
HandVu	2	3	3	3	3	14

**Tabla 2.1.** Tabla de Comparación de las herramientas existentes.

En cuanto a la naturalidad de uso que proveen las interfaces desarrolladas con estas herramientas se puede mencionar que la que mayor goza de estas propiedades son los desarrollados empleando la tecnología FTIR. Entre las tecnologías planteadas existen dos que toman como base esta técnica, TouchLib y Bespoke 3DUI XNA. En ambos casos la interacción se realiza por medio de las manos sin necesidad de elementos agregados a estas. En su versión inicial el computador sensaba el momento en el que el usuario realizaba un contacto con la superficie sensitiva para de esta manera dependiendo del lugar donde este presionaba o en base a los gestos que realizaba generar una retroalimentación. Actualmente se analiza la intensidad de la luz reflejada para poder dar una retroalimentación de navegación para momentos en los que el usuario no hace contacto con la superficie sensitiva. Por otro lado, HandVu también permite la interacción dactilológica sin el uso de elementos adicionales sin embargo, ya que se va a emplear posturas dactilológicas predeterminadas, el usuario debe conocer el lenguaje dactilológico disponible para así poder realizar una interacción más fluida. La propuesta de Lee para el uso del Wiimote para sensar la interacción por medio de las manos fue puesta en último lugar bajo este contexto debido que para realizar una detección mas precisa de la posición de los dedos del usuario, se propone

colocar cinta reflectiva en la yema de los dedos de este lo cual introduce un elemento no natural a la interfaz.

En lo relacionado a facilidad de implementación existe un balance en las alternativas presentadas. Las alternativas que emplean el Wiimote, la de Lee y Bespoke 3DUI XNA, simplifican la detección ya que esta es realizada desde hardware, por otro lado la programación para el reconocimiento de gestos es un poco mas compleja. HandVu, realiza la detección usando una cámara web y un algoritmo modificado para su correcto funcionamiento en tiempo real. Aunque su velocidad de detección y seguimiento es mayor que la de las opciones que emplean el Wiimote, la facilidad que brinda para el reconocimiento de posturas dactilológicas hace que la programación para el reconocimiento de gestos sea mas sencilla.

El rendimiento de estas herramientas se encuentra relacionada a la facilidad de implementación, ya que mientras algunas requieren menor procesamiento para la detección de puntos de contacto o posturas dactilológicas, requieren mayor procesamiento para el reconocimiento de gestos. Adicionalmente el rendimiento de estas aplicaciones se ve inversamente afectado por las dimensiones del área a monitorear en el área de trabajo. Esto es mientras mayor es el área de trabajo a monitorear, menor es el rendimiento obtenido por el sistema.

Entre los requerimientos no funcionales se encuentran el desarrollo de un sistema portable. En este contexto la alternativa

que tiene la ventaja es HandVu ya que se encuentra implementada utilizando lenguaje C++ que puede correr bajo cualquier ambiente operativo, depende de la librería OpenCV liberada bajo la licencia BSD, adicionalmente se puede compilar aplicaciones de prueba para diferentes sistemas operativos. TouchLib y la propuesta de Lee son alternativas que inicialmente fueron propuestas para funcionar bajo un entorno MS Windows sin embargo, en los últimos meses han sido experimentadas bajo otros sistemas operativos. Finalmente Bspoke 3DUI XNA se postula como la alternativa con menor portabilidad debido a que funciona estrictamente bajo ambientes Windows y depende del conjunto de clases del framework .NET, lo que hace que su migración a otros ambientes operativos sea más compleja.

Los hacks de wiimote propuestos por Johnny Lee junto con Bspoke 3DUI XNA poseen una gran desventaja en lo relacionado a independencia de hardware ya que ambas tienen un medio de entrada único, el wiimote. En este contexto HandVu encabeza la lista de alternativas ya que puede emplear tanto una webcam como una cámara DV como medio de entrada y cualquiera de los medios de salida antes mencionados para la implementación del área de trabajo. Touchlib y las tecnologías que emplean FTIR para la detección de los puntos de contacto tienen la limitante de que tienen como base el uso de proyección posterior o inferior para la implementación de la superficie sensitiva.

## **Conclusiones**

En este capítulo se presentó el alcance y las limitaciones que la solución a proponer en este documento va a poseer. Se realizó una descripción general de los componentes del sistema junto con el funcionamiento del mismo. Se mostró además la forma en la que el usuario interactuará con el mismo, por medio de sus manos, haciendo énfasis en las ventajas que esta forma de interacción posee. Finalmente, se mostraron las tecnologías disponibles para el desarrollo del sistema propuesto junto con una análisis de las mismas.

# CAPÍTULO III.

## DISEÑO

### **Diseño de Hardware**

Los equipos de hardware son una parte importante en el sistema propuesto, se encargan de monitorear las actividades realizadas por el usuario y tácitamente comprenden el medio de entrada de instrucciones del mismo. Además proporcionan la debida retroalimentación al usuario dependiendo de las acciones realizadas por el mismo. La solución que propone este documento incluye dos restricciones de diseño relacionadas al hardware: bajo costo y facilidad de implementación.

Para lograr esto, la primera decisión de diseño que se tomó en lo referente a la plataforma de hardware sobre la cual va a correr el sistema es que esta va a seguir un esquema similar al empleado en aplicaciones de visión por computador. Este esquema

comprende el uso de una cámara como medio de entrada. Esta cámara se encargará de monitorear la región de interés (ROI) con el objetivo de capturar patrones, actividades o acontecimientos según la naturaleza del problema. Para el caso particular de este sistema, la ROI a monitorear es el área de trabajo y la cámara se encargará de constantemente capturar imágenes con el objetivo de detectar las manos del usuario. Es este esquema el que nos permitirá la reutilización de hardware para el dispositivo de captura ya que al sensar las instrucciones ingresadas por un lenguaje dactilológico empleando una cámara se evita la utilización de hardware que emule una superficie sensible al tacto como es el caso de los computadores de panel o en sistemas como los ya mencionados DiamondTouch, iWindow y Tactapad. Adicionalmente se propone que el hardware a emplear para la implementación de cada uno de los componentes previamente mencionados cumpla las siguientes características:

1. Poseer un bajo costo relativo.
2. Evitar el uso de hardware especializado.
3. No verse ligado a un dispositivo de hardware determinado.

La primera característica persigue el objetivo general de esta tesis y es parte fundamental del sistema. Tanto el hardware como el software que se emplee para el desarrollo de este sistema debe minimizar los costos al máximo. La segunda característica está cercanamente relacionada con la primera, ya que por lo general mientras mas especializado es el hardware, mayor es su costo.

Otra característica del hardware especializado es que tiene menor disponibilidad, lo cual incrementa su costo. La tercera característica se encuentra un tanto relacionada con la facilidad de implementación del sistema. Al no verse ligada la plataforma de hardware a dispositivos específicos, se tiene un sistema cuyos componentes de hardware pueden ser cambiados con facilidad, lo cual nos brinda dos ventajas. La primera, está relacionada con la disminución de gastos mediante la reutilización de equipos disponibles en nuestro inventario; esto nos brinda la posibilidad de utilizar un proyector, un LCD o un TV-CRT para la implementación del área de trabajo y/o una cámara web o una cámara DV para emplearla como dispositivo de captura. La segunda, está relacionada con la fase de pruebas y actualización del sistema. Esto permite que se puedan realizar pruebas con una variedad de hardware y poder mejorar el rendimiento del sistema. Además, con el avance de la tecnología, hardware potente que en la actualidad posee un costo elevado de adquisición con el paso del tiempo experimentará un decremento en el mismo; esto hará que dicho dispositivo pueda ser integrado posteriormente al sistema, obteniendo una plataforma de hardware adaptable para futuras mejoras.

Específicamente por cada uno de los dispositivos de hardware a emplear para la implementación de cada uno de los componentes del sistema, las siguientes características deben cumplirse.

**Dispositivo de Captura:** El dispositivo de captura debe proveer una capacidad de adquisición de imágenes igual o mayor a 25 fps. El objetivo de obtener una frecuencia de captura cercana a 25 fps es sensor movimiento continuo sobre el área. Aún cuando ciertos movimientos ocurren a mayor velocidad, lo cual podría ser capturado con cámaras de alta velocidad, se persigue realizar un monitoreo a una velocidad razonable con el uso de hardware corriente.

**Área de Trabajo:** Las dimensiones de la imagen presentada en el área de trabajo dependen del dispositivo empleado para su implementación. Dependiendo de estas dimensiones el sistema proveerá una plataforma de hardware que permitirá alojar aplicaciones de software de diferentes tipos. Por ejemplo, en caso de poseer un área de dimensiones significativas en el sistema podrán correr aplicaciones en las cuales la ubicación de las posturas y gestos dactilológicos pueda ser capturada. Por otro lado, si las dimensiones del área de trabajo son menores, lo suficiente para cubrir la mano del usuario, en el sistema podrían correr sistemas controlados por posturas mas no gestos dactilológicos.

**Computador:** Uno de los requerimientos no funcionales y una de las características principales que debe tener un sistema interactivo es su respuesta en tiempo real. Por este motivo el computador a emplear no solo debe poseer las interfaces de hardware necesarias para conectar los dispositivos empleados

para la implementación del área de trabajo y del dispositivo de captura, sino también debe poseer una capacidad de procesamiento que permita el funcionamiento del sistema en tiempo real.

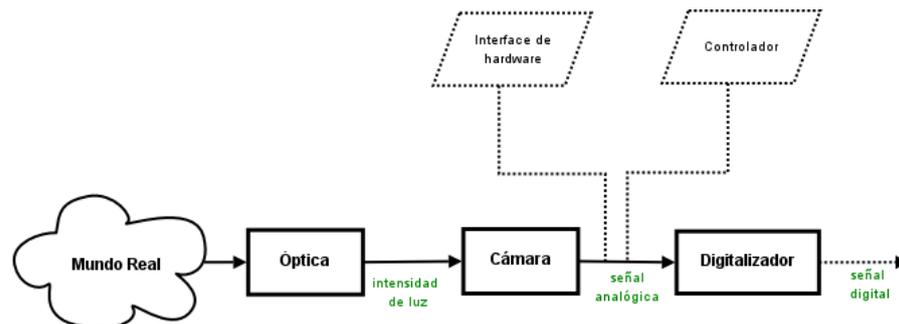
Estas son las características que el diseño de la solución propuesta va a tener en su entorno de hardware, sin embargo, para asegurar el correcto funcionamiento de este se debe establecer un diseño de software que permita la reutilización e independencia de hardware.

### **Módulos del Sistema**

En lo relacionado con su plataforma de software, el sistema propuesto está formado por los siguientes módulos.

***Módulo de Adquisición de Imágenes (MAI)*** : Este módulo se encarga de manejar la conexión con el dispositivo de captura. Es en este módulo donde se implementará la compatibilidad con diferentes dispositivos para la adquisición de imágenes, entre estos: cámaras web, cámaras DV y otros tipos de cámaras. Esto comprende la conexión e intercambio de datos. Para realizar la conexión al dispositivo de adquisición de imágenes este módulo se encarga de administrar la conexión con las diferentes interfaces de hardware entre ellas USB 1.0, USB 2.0, las variantes de IEEE1394 y futuras interfaces que mejoren la velocidad de transferencia de datos. Para manejar el intercambio de datos, este módulo administra la comunicación con los controladores

necesarios para adquirir datos desde el dispositivo empleado como dispositivo de captura. Considerando las manos del usuario en el mundo real como la fuente de los datos, el flujo de información en este módulo se da como está representado en el gráfico 3.1.



**Gráfico 3.1** Flujo de Funcionamiento del MAI.

Partiendo el Mundo Real como la fuente de datos, ya que en este se va a encontrar el área de trabajo y es el que va a ser monitoreado por el dispositivo de captura, la información referente a la escena va a pasar inicialmente por la óptica del dispositivo de captura la cual en ciertos dispositivos puede realizar operaciones de filtro o de establecimiento de ROI por medio de enfoque manual. La óptica envía información referente al nivel de intensidad de luz captada en la escena, este nivel de intensidad es el que lleva información referente al color a la cámara. En la cámara los niveles de intensidad son transformados a una señal analógica. La cámara se encuentra conectada al computador por una interfase de hardware, la misma que ofrecerá ciertas ventajas en el momento de la transferencia de datos. Para manejar la transferencia de datos con el dispositivo de captura el MAI utiliza

un programa controlador específico del dispositivo de captura empleado. Generalmente, tanto la conexión con el dispositivo como la transferencia de datos con este son manejadas por el sistema operativo. En el computador esta señal analógica es muestreada y transformada en una señal digital que representa a una imagen en dos dimensiones.

**Módulo de Detección y Seguimiento (MDS)** : La tarea de este módulo es la detección de las manos del usuario y el seguimiento de las mismas. Este módulo se implementará empleando una de las alternativas previamente mencionadas por medio de las cuales se realizará el reconocimiento de posturas dactilológicas. Como se puede observar en el gráfico 3.2 , durante su fase inicial este módulo se alimenta de la señal digital (imagen) enviada por el MAI, emplea algoritmos para la detección, seguimiento de la mano del usuario y reconocimiento de posturas dactilológicas. Como salida este módulo envía información referente al identificador de la postura reconocida como sus coordenadas de ubicación en el área de trabajo.



Gráfico 3.2. Flujo de funcionamiento del MDS.

**Módulo de Reconocimiento de Gestos (MRG)** : Este módulo se encarga de interpretar las secuencias de posturas dactilológicas

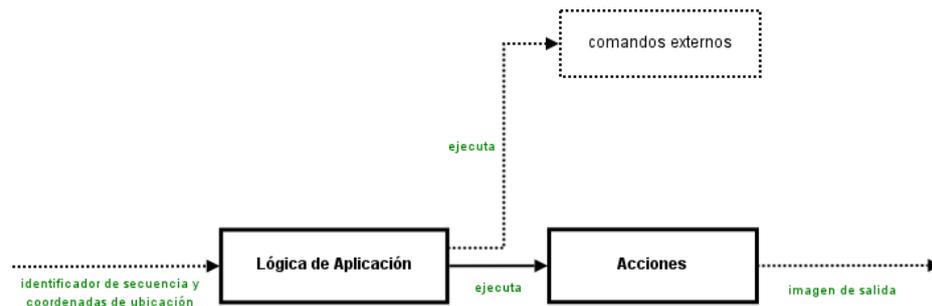
realizadas por el usuario durante la interacción con el sistema. Aquí se especifican los gestos dactilológicos empleados por la aplicación. La programación de esta capa está ligada a la aplicación a correr en el sistema. Para lograr la interpretación de las secuencias de posturas dactilológicas realizadas por el usuario, como se muestra en la gráfica 3.3, este módulo recibe del MDS los identificadores de cada una de las posturas dactilológicas reconocidas por este. Dependiendo de la aplicación a correr en el sistema, un conjunto de secuencias serán reconocidas representando cada una de estas secuencias a un gesto dactilológico. Cuando uno de estos gestos es reconocido la información referente al identificador del gesto o a la acción a realizar como las coordenadas de ubicación del mismo son enviados como salida al siguiente módulo.



**Gráfico 3.3.** Flujo de Funcionamiento del MRG.

**Módulo de Aplicación (MA) :** En este módulo se especifica la lógica de la aplicación. Al igual que el MRG la programación de esta capa está ligada a la aplicación a correr en el sistema. Como se puede ver en el gráfico 3.4, este módulo recibe como datos de entrada los identificadores y coordenadas de ubicación de los gestos dactilológicos reconocidos por el MRG. Dependiendo de la

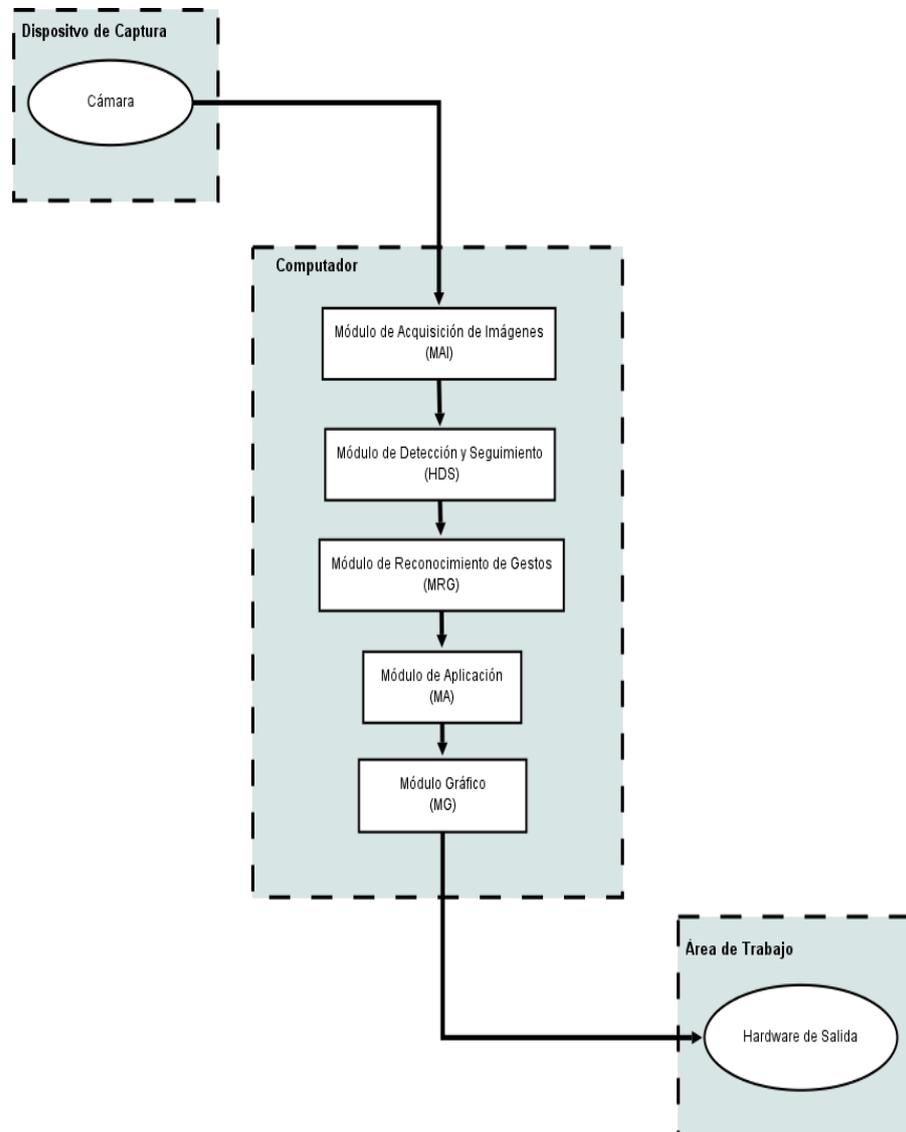
aplicación y de la información recibida referente a los gestos reconocidos, este módulo realizará ciertas acciones o ejecutará comandos. Estas acciones y/o comandos se encargaran de generar la imagen que luego va a ser presentada en el área de trabajo.



**Gráfico 3.4.** Flujo de Funcionamiento del MA.

**Módulo Gráfico (MG) :** El módulo gráfico se encarga de manejar la GUI mostrada en el área de trabajo y de proveer retroalimentación al usuario relacionada a las instrucciones que este realice. Esta librería utiliza la imagen generada por el MA, junto con librerías gráficas, librerías de sonido, librerías para diseño de entornos gráficos, motores de física, etc se encarga de enviar una retroalimentación al usuario dependiendo a las acciones realizadas por este sobre el área de trabajo.

De esta manera, el flujo de datos entre componentes y módulos del sistema puede resumirse en el gráfico 3.5.



**Gráfico 3.5.** Módulos del Sistema propuesto.

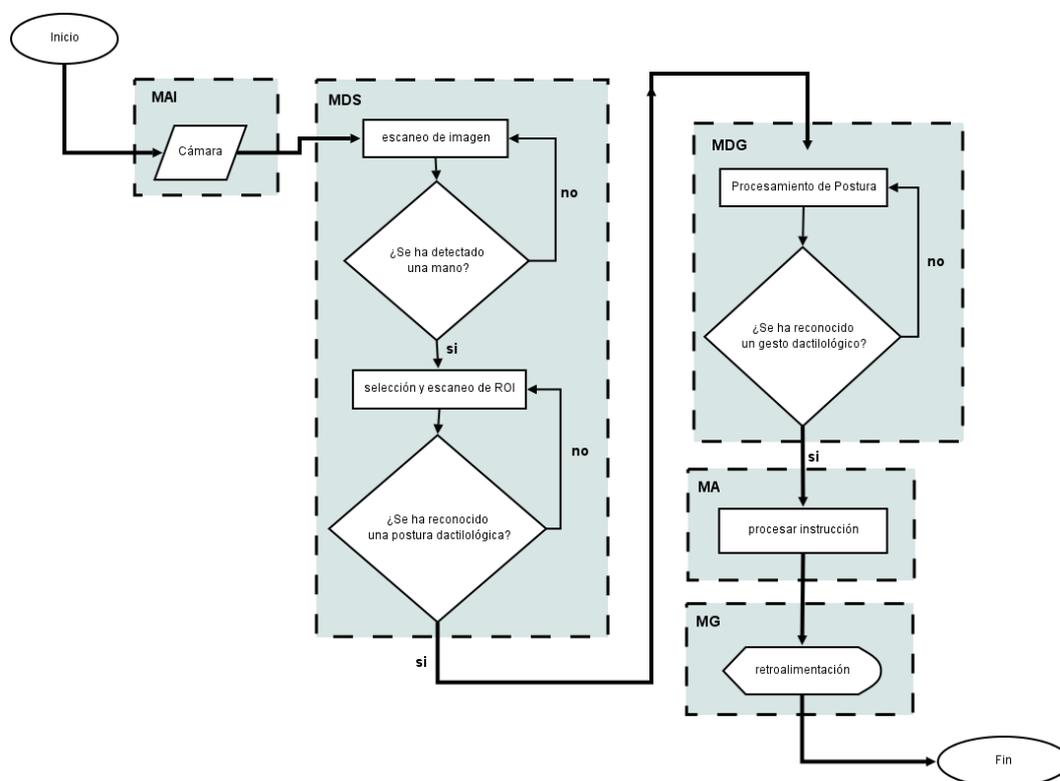
### **Funcionamiento e Interacción entre Módulos del Sistema**

Para realizar lo expuesto anteriormente, y brindar la funcionalidad especificada el sistema sigue el siguiente procesamiento:

En la primera etapa del procesamiento, cuando el usuario interactúa sobre el área de trabajo, una cámara apuntando perpendicularmente al área de trabajo constantemente captura

imágenes de la misma. Esta cámara es controlada desde el MAI. Estas imágenes son enviadas al MDS, módulo en el cual se analiza la imagen recibida para realizar la detección de la mano del usuario. Para realizar esta detección, el sistema establece una ROI sobre la imagen. En esta ROI el usuario deberá colocar su mano para poder iniciar la interacción con el sistema. Una vez que la mano del usuario es localizada, esta es seguida y el área cercana a esta es escaneada con el objetivo de identificar posturas dactilológicas. La información relacionada a localización y postura de la mano del usuario es enviada al MRG. Esta información es empleada para la interpretar las secuencias de posturas dactilológicas realizadas por el usuario de forma que sean empleadas como medio de ingreso de instrucciones en la aplicación. Adicionalmente, este módulo realiza un registro de las posturas reconocidas por el MDS empleado con fines experimentales. El MA es específico a la funcionalidad que se desee del sistema y maneja la lógica de la aplicación. En el caso de la aplicación de pruebas a desarrollar el MA manejaría la lógica del procesamiento de imágenes. Finalmente, el MA envía retroalimentación al usuario por medio del MG quien administra la GUI mostrada en el área de trabajo.

El gráfico 3.6 presenta brevemente el funcionamiento e interacción entre los módulos que conforman el sistema.



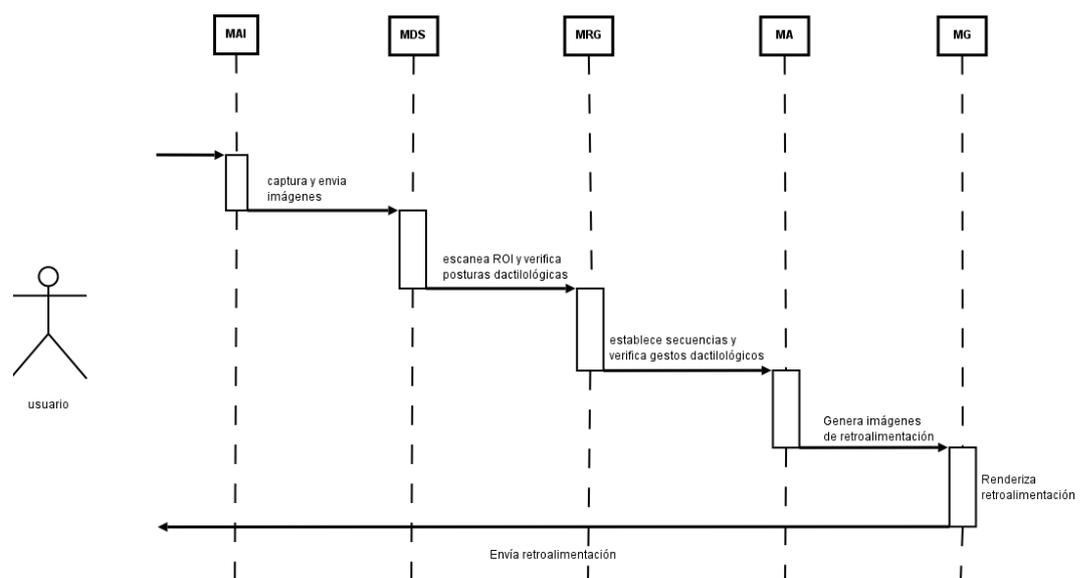
**Gráfico 3.6.** Interacción entre los Módulos del Sistema.

Para demostrar de mejor manera la interacción entre los módulos del sistema, se presentará Diagramas de Secuencia de cuatro escenarios comunes.

**a) Escenario:** Usuario realiza una operación correctamente

**Descripción:** El usuario realiza una de las operaciones antes mencionadas, inicialmente por medio de una postura y luego por medio de un gesto dactilológico. El MIA envía las imágenes del área de trabajo que capturó al MDS. El MDS establece una ROI en las imágenes recibidas y la escanea para el reconocimiento de

posturas dactilológicas. Al reconocer una postura envía el evento al MRG quien verifica la secuencia en la que se han realizado las posturas para reconocer gestos dactilológicos. Al detectar una secuencia relacionada a una operación, el grado en que el gesto es realizado es enviado al MA quien ejecuta la operación sobre las imágenes en el mismo grado en el que fue realizado el gesto. Finalmente el MG recibe la imagen generada después de la ejecución de la operación y la presenta en el área de trabajo. En el gráfico 3.7 se muestra el diagrama de secuencia que describe el procesamiento en dicho escenario



**Gráfico 3.8.** Diagrama de Secuencia de la correcta ejecución de una operación

**b) Escenario:** Usuario inicia la interacción con el sistema.

**Descripción:** El usuario coloca su mano sobre el área de trabajo para dar inicio a la interacción con el sistema. El MIA captura y envía las imágenes del área de trabajo al MDS. El MDS establece una ROI en las imágenes recibidas y la escanea para el reconocimiento de posturas dactilológicas. Al reconocer la postura

inicial envía el evento al MRG quien verifica la secuencia en la que se han realizado las posturas para reconocer gestos dactilológicos. Al no detectar una secuencia predeterminada, sede el procesamiento al MA sin evento que notificar. Al no haber una secuencia que describa el gesto se genera una imagen de movimiento. Finalmente el MG recibe la imagen generada y la presenta en el área de trabajo.

En el gráfico 3.8 se muestra el diagrama de secuencia que describe el procesamiento en dicho escenario

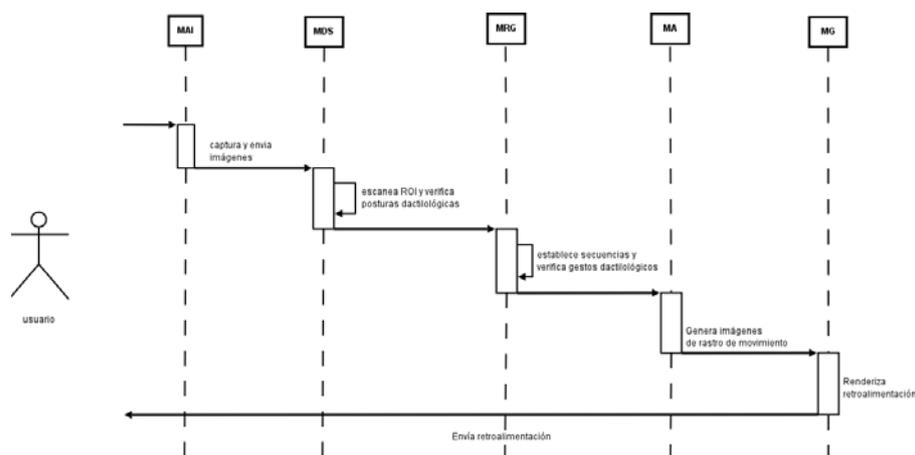


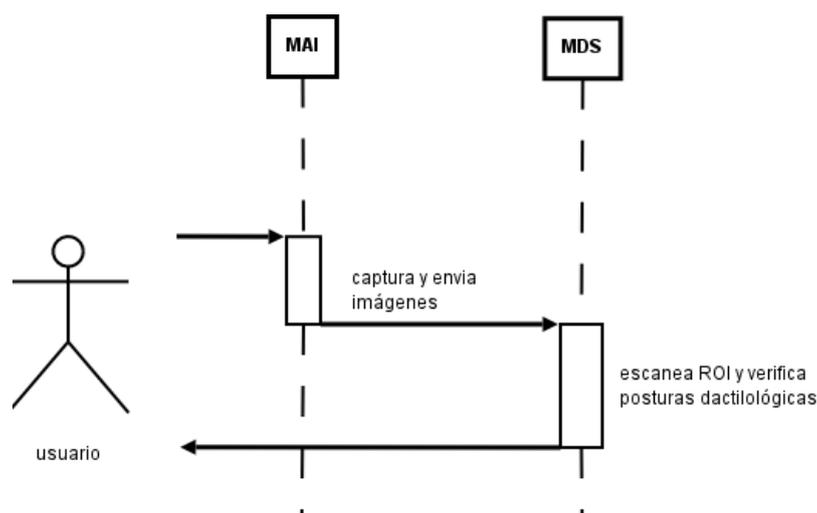
Gráfico 3.8. Diagrama de Secuencia del inicio de interacción con el sistema.

**c) Escenario:** Postura dactilológica realizada por el usuario no es reconocida por el sistema.

**Descripción:** El usuario desea realizar una de las operaciones antes mencionadas, inicialmente por medio de una postura y luego por medio de un gesto dactilológico. El MIA envía las imágenes del área de trabajo que capturó al MDS. El MDS establece una ROI en las imágenes recibidas y la escanea para el reconocimiento de posturas dactilológicas. El MDS no reconoce

postura alguna en la ROI escaneada por lo que repite el proceso de establecimiento de ROI y escaneo hasta reconocer una postura y pasar el proceso al MRG.

En el gráfico 3.9 se muestra el diagrama de secuencia que describe el procesamiento en dicho escenario



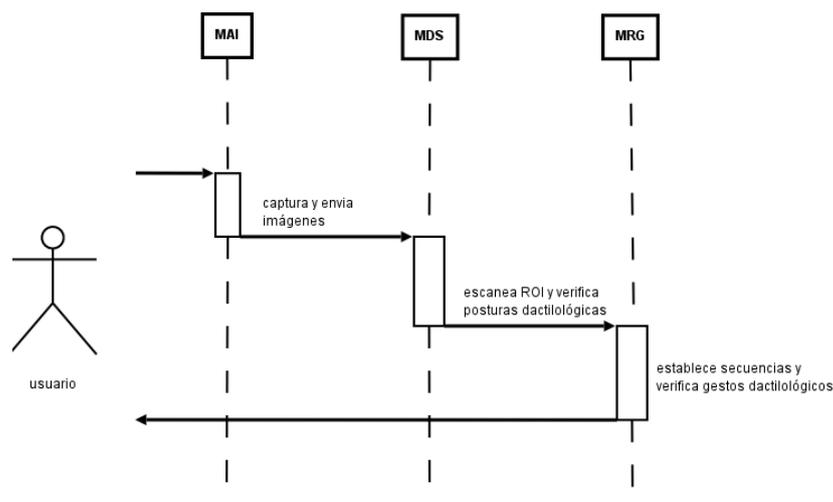
**Gráfico 3.9.** Diagrama de Secuencia del proceso en el cual la postura dactilológica realizada por el usuario no es reconocida.

**d) Escenario:** *Gesto dactilológico realizado por el usuario no es reconocido por el sistema.*

**Descripción:** El usuario desea realizar una de las operaciones antes mencionadas, inicialmente por medio de una postura y luego por medio de un gesto dactilológico. El MIA envía las imágenes del área de trabajo que capturó al MDS. El MDS establece una ROI en las imágenes recibidas y la escanea para el reconocimiento de posturas dactilológicas. Al reconocer una postura envía el evento al MRG quien verifica la secuencia en la que se han realizado las posturas para reconocer gestos dactilológicos. Al no detectar una secuencia relacionada a una

operación, el MRG se mantiene procesando nuevos eventos recibidos por el MDS. La secuencia de posturas que se mantenía y el área de trabajo se mantienen sin cambios.

El gráfico 3.10 muestra el diagrama de secuencia que describe el procesamiento en dicho escenario



**Gráfico 3.10.** Diagrama de Secuencia del proceso en el cual el gesto dactilológico realizado por el usuario no es reconocido.

## Arquitectura del Sistema

Según lo expuesto previamente se puede evidenciar que al ser una aplicación standalone el sistema propuesto sigue una arquitectura de capas o también conocida como modelo de máquina abstracta (12). Aún cuando entre las alternativas existentes para el desarrollo del sistema algunas proveen la característica de funcionar bajo una arquitectura Cliente-Servidor, como es el caso de Touchlib y HandVu. El sistema a desarrollar no se registrará bajo este esquema ya que la arquitectura Cliente-Servidor implica el uso de computadores adicionales, el desarrollo

de programas o módulos clientes, manejo de comunicaciones y demás requerimientos que atentan contra el bajo costo y fácil implementación que el diseño del sistema a proponer persigue. Bajo la arquitectura de capas cada uno de los módulos mencionados anteriormente constituirían las capas del sistema. De esta manera cada capa recibirá un servicio de la capa inferior y proveerá servicios a la capa superior. La arquitectura de capas provee ciertas ventajas al sistema, entre ellas:

- ***Fácil Modificación y Portabilidad:*** La arquitectura por capas permite que en caso de desear realizar alguna actualización o agregar funcionalidad adicional a una de las capas, la modificación sea realizada solo a la capa en cuestión y no al sistema como un todo. Esto facilita la portabilidad de la aplicación y la liberación de versiones multiplataforma del sistema por medio de la modificación de capas que se encuentren estrechamente ligadas al sistema operativo en el que corran.
- ***Soporte de desarrollo Incremental:*** Por medio de esta arquitectura el sistema dará la facilidad de que a medida de que cada capa es implementada pueda ser colocada en un proceso de pruebas. Con esto se garantiza la funcionalidad de cada capa sin la necesidad de que el sistema sea desarrollado en su totalidad.

## **Diseño de Software**

Como fue planteado en el capítulo de Análisis el sistema a desarrollar debe permitir su portabilidad a diferentes sistemas operativos. Esto implica que las librerías, marcos de trabajo y demás herramientas empleadas para la implementación de los módulos deben ligar la implementación inicial al sistema operativo lo menos posible. Implícitamente se plantea que herramientas que solo funcionen bajo un ambiente de software en particular deben ser ignoradas. Sin embargo, gracias a la arquitectura por capas empleada en el diseño del sistema, dichos módulos implementados con herramientas específicas de un sistema operativo pueden ser modificadas posteriormente. Otro requerimiento no funcional que debe ser tomado en cuenta, es el relativo bajo costo que la implementación del sistema debe poseer. Por este motivo, se va a dar prioridad a la revisión y prueba de herramientas libres y de código abierto que se encuentren libre de costos, para de esta manera evitar incurrir en el gasto de licencias tanto para la herramienta como para los entornos de software en los que estas correrán.

## **Modelo de Datos**

Como se presentó en la sección de limitaciones del sistema, en el capítulo de Análisis, ya que el objetivo de la aplicación de manipulación de imágenes es probar la interacción con el tipo de interfase a desarrollar, el sistema no manejará persistencia de

datos. En la aplicación de pruebas la información referente a la manipulación realizada por el usuario se manejará de forma temporal. Cada vez que el usuario inicie la aplicación un número de imágenes predeterminadas se cargaran en ubicaciones aleatorias. Existe un conjunto de clases para el manejo de la información referente a la ubicación, escala y ángulo de rotación de las imágenes cargadas sin embargo esta información se mantendrá en memoria mientras la aplicación se mantenga corriendo.

Para la lógica de la aplicación de manipulación de imágenes, se establecieron las siguientes clases:

***vtDesktop:*** Esta clase es la que representa al área de trabajo para esta aplicación. Contiene información relacionada con la imagen a mostrar en el área de trabajo, los archivos de imágenes cargados en la aplicación, el nivel de profundidad de las imágenes colocadas en el área de trabajo. Entre los métodos y funciones con las que cuenta, provee el renderizado de la pantalla, la colocación de imágenes aleatorias, entre otras.

***vtZbuffer:*** Esta clase sirve para representar la vista directa del área de trabajo. Es la unión de todas las capas en las que se cargaron los diferentes archivos de imágenes. En la lógica de la aplicación es empleada para poder llevar un seguimiento de las regiones que ocupa cada una de las imágenes dentro del área visible.

***vtImage***: Esta es la clase que manejará la información de los archivos cargados durante la ejecución de la aplicación. Entre los datos que contiene se puede mencionar ubicación, escala, ángulo de rotación, área mostrada, URL, etc. . Esta clase contiene los métodos y funciones necesarias para la traslación, rotación y cambio de tamaño de las imágenes en la aplicación.

## **Conclusiones**

En este capítulo se presentaron las características del producto que el diseño del sistema propuesto persigue. En cuanto a hardware, promueve la independencia de hardware; disminuyendo los costos permitiendo el uso de diferentes tipos de hardware. En cuanto a software, plantea el desarrollo de un sistema portable, que evite tomar como base de desarrollo librerías nativas del entorno en el que corre para así poder tener un sistema multiplataforma. En cuanto a especificaciones del procesamiento de software, se presentaron los módulos con los que la solución propuesta cuenta y por medio de diagramas de secuencia se presentó el flujo de trabajo y la comunicación entre dichos módulos en diferentes escenarios propuestos

# CAPÍTULO IV.

## IMPLEMENTACIÓN

Para la implementación de la solución propuesta se va a considerar lo planteado previamente en los capítulos de Análisis y Diseño. El hardware empleado va a seguir los lineamientos planteados en la sección de Diseño de Hardware mientras que se para el desarrollo del sistema, se va a considerar el análisis realizado previamente referente a las alternativas existentes para la detección, seguimiento y reconocimiento de posturas y gestos dactilológicos.

### **Tecnología**

### **Sistema Operativo**

Para liberar tanto a los desarrolladores como a los usuarios finales del sistema de la necesidad de adquirir licencias de software, la plataforma de software sobre la que el sistema

propuesto va a correr va a ser GNU/Linux. Este sistema operativo fue seleccionado, adicionalmente de estar libre de costos, por la libertad con la que se cuenta para modificar las aplicaciones y librerías bajo la Licencia General Pública (GPL) (9), licencia que cubre a las aplicaciones libres. Otro de los criterios por los cuales se seleccionó este sistema operativo es por el alto rendimiento y estabilidad que brinda, esto da la posibilidad de correr la aplicación sobre hardware que no necesariamente sea reciente o potente.

### **Lenguaje de Programación**

El lenguaje en el que se va a desarrollar el sistema propuesto va a ser C++. Este lenguaje de programación fue seleccionado por razones de rendimiento y compatibilidad. En lo relacionado a rendimiento, considerando que uno de los requerimientos no funcionales es que la aplicación sea apta para responder en tiempo real; las aplicaciones desarrolladas en este lenguaje poseen un funcionamiento independiente de máquinas virtuales como en el caso de las aplicaciones escritas en lenguajes como Java que depende de la Máquina Virtual de Java, lo cual incrementa una capa de procesamiento afectando el rendimiento y por ende el tiempo de respuesta. En lo relacionado a compatibilidad, ya que HandVu se postula a ser la herramienta a emplearse para reconocimiento de posturas y gestos dactilológicos, y ya que la aplicación de procesamiento de

imágenes podría ser desarrollada en base a OpenCV, ambas librerías escritas en C++, el uso de este lenguaje permitiría fácil compatibilidad entre la parte de software que se encarga del procesamiento del lenguaje dactilológico con la parte encargada de la aplicación.

## **Implementación de los Componentes del sistema**

### **Componentes de Hardware**

**Dispositivo de Captura:** Para la implementación del dispositivo de captura se consideró tres cámaras web que estaban disponibles. El uso de estas cámaras para la implementación del sistema confirmaría la independencia de hardware en lo referente al componente de dispositivo de captura. Estas cámaras son la Logitech Quickcam Express, Creative NX y Genius Messenger, con una resolución de 320x240, 640x480 y 640x480 píxeles, respectivamente.

A continuación, en la tabla 4.1, se muestran las especificaciones de estas cámaras.

Especificaciones	Imagen
<p><b>Logitech Quickcam Express</b></p> <p>Resolución: 320x240</p> <p>Profundidad de color: 24 bit</p> <p>Conector: USB 1.0</p> <p>frecuencia de captura: 30 fps</p> <p>sensor de imagen: CMOS</p> <p>tipo de sistema: PC (Intel x86)</p>	
<p><b>Creative NX Ultra</b></p> <p>Resolución: 640x480 300 k píxeles</p> <p>Profundidad de color: 32 bits</p> <p>Conector: USB 1.1</p> <p>Frecuencia de captura: 30 fps</p> <p>Sensor de imagen: CCD</p>	
<p><b>Genius Messenger</b></p> <p>Resolución: 640x480 300 k píxeles</p> <p>Profundidad de color: 32 bits</p> <p>Conector: USB 1.1</p> <p>Control de Exposición: automático.</p> <p>Frecuencia de captura: 30 fps</p> <p>Sensor de imagen: CMOS</p>	

**Tabla 4.1.** Tabla de cámaras usadas para la implementación del dispositivo de captura

**Área de Trabajo:** En lo que corresponde a este componente. Se empleó un monitor LCD de 24 pulgadas. Adicionalmente, se implementaron prototipos de prueba utilizando una pantalla de retroproyección, y un televisor CRT de 21 pulgadas, para de esta manera verificar el funcionamiento del sistema empleando diferentes dispositivos de salida. Tomando en cuenta que un gesto dactilológico es interpretado como una secuencia de ubicaciones de la mano, para la implementación de este componente se recomienda el uso de dispositivos que provean de un área de proyección mayor a 21 pulgadas. De esta forma el sistema contará con un Área de Trabajo de dimensiones considerables para poder realizar un desplazamiento. Para aplicaciones que únicamente tomen como entrada posturas dactilológicas independientemente de la ubicación de estas, se puede emplear cualquier dispositivo sin tener restricciones en las dimensiones de estos. En la tabla 4.2 se puede observar las especificaciones de los equipos empleados para la implementación de este componente.

Especificaciones	Imagen
<p><b>Sony KV-20S42 TV</b></p> <p>Tipo: Tubo de Rayos Catódicos</p> <p>Dimensiones: 22 Pulgadas</p> <p>Ratio de Aspecto: 4:3</p> <p>Resolución: 800x600</p>	
<p><b>SUN 24.1 TFT</b></p> <p>Tipo: Pantalla de Cristal Líquido</p> <p>Dimensiones: 24.1 Pulgadas</p> <p>Ratio de Aspecto: 16:10</p> <p>Resolución: 1920x1200</p>	
<p><b>Lámina translúcida para retroproyección.</b></p> <p>Dimensiones: 42 Pulgadas</p> <p>Ratio de Aspecto: 4:3</p> <p>Resolución: 800x600</p>	

**Tabla 4.2.** Dispositivos empleados para la implementación del Área de Trabajo.

**Computador:** El computador es uno de los componentes clave en el cumplimiento del requerimiento de funcionamiento en tiempo real. Dependiendo del poder de procesamiento de este, la captura de imágenes podrá ser realizada con una frecuencia adecuada. El reconocimiento del lenguaje dactilológico se podrá realizar en un

tiempo relativamente bajo haciendo que la aplicación pueda generar una retroalimentación en tiempo real. Durante su desarrollo el sistema fue probado en un computador de escritorio con procesador Intel Pentium 4 de 1.8 GHZ con 1 GB de memoria y en un computador portátil con procesador dual core de 1.6 GHZ y 2GB de memoria. En ambos casos el rendimiento fue fluido sin embargo para el proceso de pruebas y debido a la facilidad de movilidad que proveía, se empleó el computador portátil para la implementación del sistema. Las especificaciones de este dispositivo se incluyen en la tabla 4.3

Especificaciones	Imagen
Procesador: Intel T2300 1.66 GHz Memoria: 2GB Video: ATI Mobility Radeon X1400 256 MB Hyperthread.	

**Tabla 4.3.** Especificaciones del computador empleado

## Componentes de Software

**Software:** Como se mencionó en la parte inicial de este capítulo el sistema se va desarrollar sobre el sistema operativo GNU/Linux; específicamente, se empleó la distribución OpenSUSE.

La selección de esta distribución se dio por dos razones:

- Es la distribución empleada como plataforma base en el Centro de Visión y Robótica de la ESPOL. Lo cual confirma su potencial para alojar este tipo de aplicaciones.
- En lo personal, OpenSUSE es una distribución de GNU/Linux empleada por más de tres años y sobre la cual se ha desarrollado anteriormente sistemas de visión por computador. La experiencia de uso y la facilidad que esta posee para la instalación de paquetes de software la hace buena para ser integrada en el sistema.

Adicional a esto se empleó Anjuta, un ambiente de desarrollo incluido en la distribución, como herramienta para el desarrollo del sistema.

Como se mostró en el capítulo anterior, el componente de software está constituido por cinco módulos. A continuación se describe la implementación de cada uno de estos módulos.

**MAI:** Como se describió en el capítulo anterior, este módulo se encarga de manejar tanto la conexión con el dispositivo de captura como la transmisión de datos desde este. Para esto se empleó el paquete Gspcav; controlador de cámaras Spca5xx, Spca5xx-LE y Zr364xx, y el paquete Uvcvideo; controlador de dispositivos USB de clase video, para manejar la conexión de estos equipos. Para manejar la comunicación con estos desde el sistema, se utilizó las funciones para el control de capturas incluidas en la librería OpenCV.

**MDS:** Como fue presentado en el capítulo de Análisis, HandVu es la herramienta que en base a las características analizadas es la mas apta para ser empleada para el desarrollo del MDS. Esta herramienta depende de OpenCV, librería empleada para el desarrollo del MAI. A continuación, en el gráfico 4.1, se presenta una aplicación de prueba de HandVu en funcionamiento.

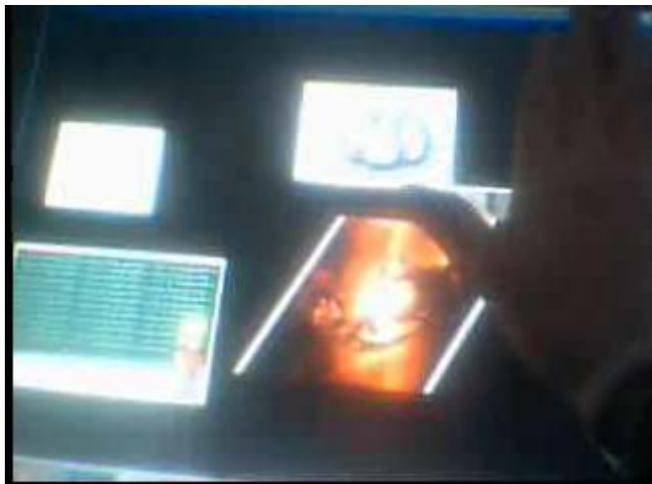


**Gráfico 4.1.** HandVu

**MRG:** Ya que en la aplicación de prueba se emplean gestos dactilológicos cuya base son los desplazamientos unidireccionales, no se empleó librería especial alguna. La implementación de este módulo utilizó librerías C estandarizadas y el manejo de secuencias de posturas de este módulo se implementó por medio de programación dinámica.

**MA:** Para la implementación de este módulo se utilizó la librería OpenCV. Esta librería contiene un conjunto de funciones para el procesamiento de imágenes la cual es empleada para el desarrollo de sistemas de procesamiento de imágenes y visión por computador. En la aplicación de prueba, esta librería se encargará de cargar las imágenes junto con su información correspondiente y de manejar la rotación, cambio de ubicación y tamaño de las mismas.

**MG:** Dadas las características de la aplicación de pruebas, esta no requiere de un ambiente gráfico tan avanzado. Por este motivo y para evitar la instalación de herramientas adicionales, se empleó HighGUI para la implementación de este módulo. HighGUI es una librería incluida en OpenCV que provee de un conjunto de funciones para el desarrollo de interfaces gráficas minimalistas como la presentada en el gráfico 4.2.



**Gráfico 4.2.** Aplicación de Manipulación de Imágenes.

En la tabla 4.3 se resume las herramientas empleadas para el desarrollo de cada uno de estos módulos:

Módulo	Herramienta
MAI	OpenCV
	gspcav1
	Uvcvideo
MDS	HandVu

**Tabla 4.4.** Herramientas empleadas para la implementación de los módulos del sistema.

## **Algoritmo de Detección, Seguimiento y Reconocimiento de posturas dactilológicas de HandVu.**

Una vez que el sistema empieza a correr, la cámara colocada en la parte superior captura imágenes a una frecuencia de entre 40 y 50 frames por segundo. Se hace una captura a una frecuencia mayor que la empleada en archivos de video, generalmente 30 frames por segundo para poder realizar el procesamiento necesario sobre estas y poder obtener resultados que puedan ser aptos para tiempo real. Esta secuencia de imágenes es enviada hacia el MDS. Como se mencionó en la sección anterior, el MDS será implementado tomando como base la librería HandVu. En el MDS, HandVu escaneará el área donde el usuario deberá colocar su mano para iniciar la detección. En el momento de la detección, el sistema aprenderá el color de piel del usuario en un histograma RGB y lo contrastará con el fondo. Luego se realiza la selección de características KLT (Kanadi-Lucas-Tomasi), esto comprende un número determinado de puntos descriptores de un objeto en particular, en este caso la mano del usuario. Una vez que las características KLT han sido seleccionadas inicia la fase de seguimiento. Durante esta fase, se verifica constantemente que los puntos de las características KLT seleccionadas se mantengan a tanto a una distancia mínima como máxima entre sí, ya que con el movimiento y cambio de postura de la mano, estos puntos tienden a cambiar de posición. En caso de que uno de

estos puntos rompa con la restricción de distancias, se selecciona otro que cumpla con dichos requerimientos. Durante la etapa de seguimiento, HandVu constantemente verifica las posiciones de las características KLT para el reconocimiento de posturas dactilológicas. En el momento que una postura es reconocida, el proceso de seguimiento se inicializa generando las características KLT en base a la postura reconocida. Esto amplía el proceso de seguimiento.

Ya que el objetivo de este documento es la implementación del sistema más que el estudio y desarrollo de un algoritmo para el reconocimiento de posturas/gestos dactilológicos, no se entrará en detalles más específicos en cuanto al funcionamiento de HandVu.

### **Procesamiento de la Aplicación de manipulación de imágenes**

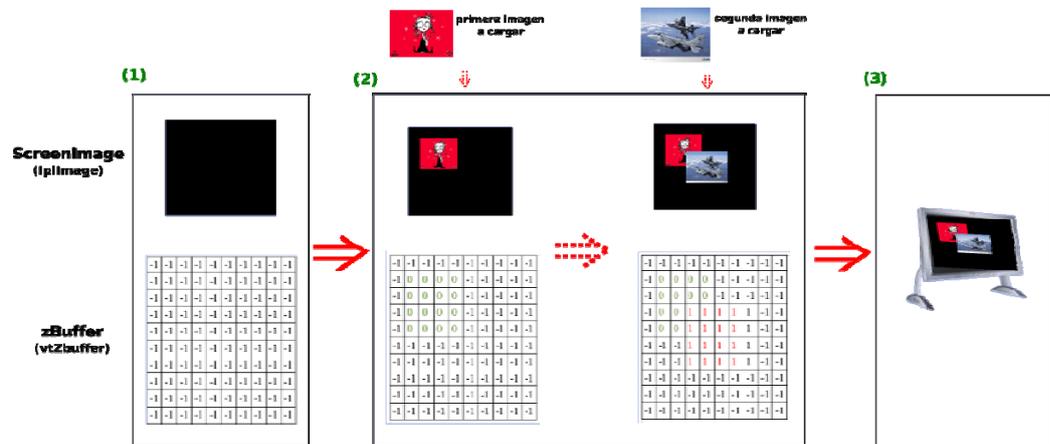
Al inicio de la ejecución de la aplicación de procesamiento de imágenes, las primeras líneas de ejecución se encargan de levantar los procesos necesarios para el funcionamiento del MAI y del MDS. Entre estos procesos la gran mayoría tienen que ver con HandVu, la librería empleada para la detección y seguimiento de posturas dactilológicas. Luego de esto, como se muestra en el gráfico 4.3, ocurre la siguiente secuencia de pasos:

1. Se instancia un objeto `IpImage` que representará a `screenImage`, la imagen mostrada en el área de trabajo. Adicionalmente, se instancia un objeto de tipo `vtZbuffer`, el

mismo que nos es otra cosa mas que un arreglo de enteros con un número de elementos igual al número de píxeles de la imagen a mostrar en el área de trabajo. Al inicio de la aplicación este arreglo es inicializado dando el valor de -1 a cada uno de sus elementos. La función del vtZbuffer es almacenar información referente a la coordenada Z de cada una de las imágenes mostradas en la aplicación. Esto sirve para llevar un control sobre el nivel de profundidad de cada imagen con respecto de las demás.

2. Por cada una de las imágenes a cargar en la aplicación:
  - 2.1. Se instancia un objeto `IpImage` para alojarla dentro de un objeto `vtImage` que contiene información adicional necesaria para la aplicación tal como: ubicación, factor de escala, ángulo de rotación, etc. Este objeto `vtImage` va a ser parte de un arreglo de elementos de este tipo contenidos en un objeto `vtDesktop` que representa el área de trabajo.
  - 2.2. Se crea una matriz que va a servir para especificar una región de interés (ROI) en la imagen principal sobre la cual va a ser colocada.
  - 2.3. La imagen a cargar es copiada en la ROI.
  - 2.4. Se actualiza el `vtZbuffer` con el valor de Z de la imagen cargada. A medida que se van cargando imágenes este valor va aumentando. De esta manera, la imagen que se encuentre sobre las demás tendrá un valor Z que tenderá a ser mayor.

- Luego de que todas las imágenes preseleccionadas han sido cargadas, se renderiza en pantalla screenImage.



**Grafico 4.3.** Flujo de Procesamiento inicial de la aplicación durante el renderizado de imágenes.

Mientras el MDS por medio de las funciones proveídas por HandVu realiza el constante escaneo de la mano del usuario para el reconocimiento de posturas dactilológicas. Cuando una postura es reconocida, la aplicación realiza el siguiente procesamiento (gráfico 4.4) :

- Se extrae la ubicación en la cual la postura dactilológica fue reconocida.
- Se verifica si la postura dactilológica reconocida es una de las posturas empleadas por el sistema para el inicio de una de las operaciones mencionadas previamente (trasladar, rotar, escalar).
- Se verifica si ya se había dado inicio a una operación previamente que no haya sido concluida.

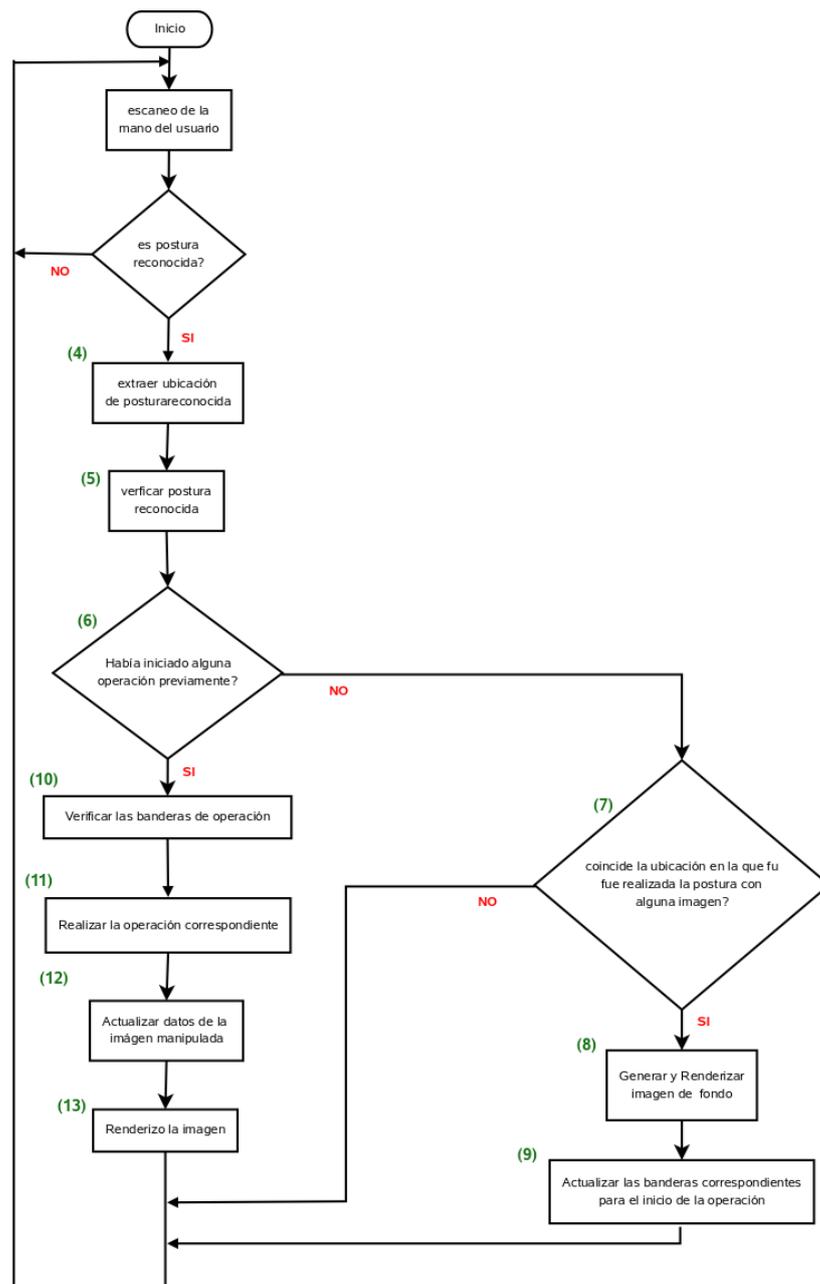
En caso de que no se haya una operación por concluir,

7. Se verifica si la ubicación en la que se realizó la postura coincide con la ubicación de alguna de las imágenes cargadas en la aplicación. En caso de coincidir, se activa la bandera `isSelected`, que indica que se inició una operación sobre una imagen.
8. Se renderiza el fondo de la imagen seleccionada
  - 8.1. Se inicializa el fondo de la imagen de fondo empleando la función `cvSetZero` proveída por OpenCV.
  - 8.2. Para cada una de las imágenes cargadas, aquellas dentro del arreglo de `vImage` dentro del `vtDesktop`, se crea una matriz y se establece una ROI para cada una de ellas dentro de la imagen de fondo. Esta ROI es útil tanto para disminuir el tiempo de procesamiento como para validar el caso en que la imagen a cargar se encuentre clipeada en alguno de sus extremos.
  - 8.3. Las imágenes que tengan un valor Z menor que el de la imagen seleccionada lo mantienen, mientras que las imágenes con valor Z mayor serán actualizadas recibiendo un decremento en una unidad en dicho valor.
  - 8.4. Se renderiza la imagen de fondo.
  - 8.5. Se crea una matriz y establezco una ROI para la imagen seleccionada.

- 8.6. Se actualiza el valor Z de la imagen seleccionada con el mayor valor Z del arreglo de imágenes. Esto hace que la imagen seleccionada se encuentre sobre las demás.
9. Dependiendo de la postura de inicio de instrucción reconocida se activa una de las banderas: `isMoving`, `isRotating` o `isScaling`, para indicar la operación que el usuario desea realizar a medida que el usuario desplace su mano sobre el área de trabajo.

Si el estado del escáner de HandVu es "En Seguimiento", se realiza el siguiente procesamiento:

10. Se verifica el tipo de operación a realizar por medio de las banderas: `isMoving`, `isRotating` e `isScaling`.
11. Dependiendo de la bandera activada se llama a la función para realizar la operación respectiva y se realiza el procesamiento interno de imágenes.
12. Se actualizan los valores de ubicación, escala y/o ángulo de rotación dependiendo de la operación realizada.
13. Se actualiza y renderiza la imagen mostrada en el área de trabajo.



**Grafico 4.4.** Flujo de Procesamiento para la manipulación de imágenes en la aplicación.

## Conclusión

En este capítulo se presentó los dispositivos de hardware empleados para la implantación del sistema. Para la selección de estos se consideró las características que el sistema debía poseer

en lo referente a hardware según lo planteado en el capítulo de diseño: relativo bajo costo, independencia de hardware.

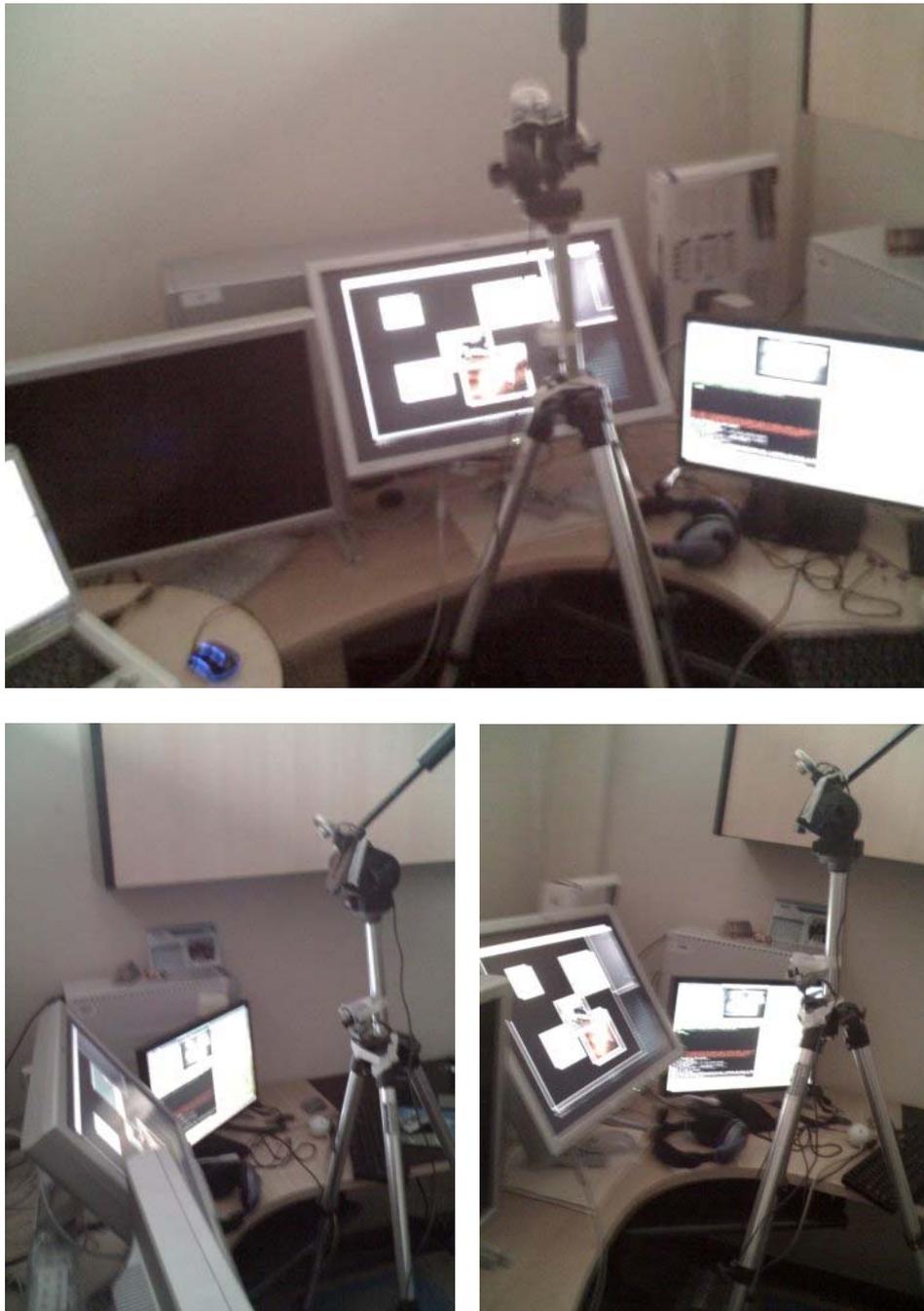
En lo referente a software, el sistema fue desarrollado sobre ambiente GNU/Linux, OpenSUSE para ser mas específicos. Se seleccionó a Anjuta como ambiente de desarrollo y para la implementación de los módulos se empleó principalmente a HandVu como núcleo del MDS, y a OpenCV para manejar la captura de imágenes como la manipulación de las imágenes manipuladas por el usuario, junto a un conjunto de controladores de dispositivos de video y librerías C estandarizadas.

Finalmente, se dio una presentación breve del algoritmo empleado por HandVu para la detección, seguimiento y reconocimiento de posturas dactilológicas, y del algoritmo utilizado dentro del MA para el funcionamiento de la aplicación.

# **CAPÍTULO V.**

## **PRUEBAS Y RESULTADOS**

Como se mencionó previamente, la solución propuesta, cuenta con varios elementos tanto en los componentes de hardware como de software. Para verificar su efectividad se realizaron cinco pruebas. Para esto se realizó una implementación con computador fijo, sin embargo el hardware empleado para los componentes de área de trabajo y dispositivo de captura fueron reemplazados durante el proceso de pruebas tanto para verificar la independencia de hardware que el sistema debe poseer como para mejorar la eficacia de este.



**Gráfico 5.1.** Implantación del sistema empleando un monitor LCD.

## **Prueba 1 (P-1)**

### **Descripción de la Prueba**

Para la selección de HandVu como base sobre la cual se implementaría el MDS del sistema, se realizó una prueba para

verificar su efectividad en la detección de las manos del usuario. Para esto se instaló la librería y se empleó la aplicación de ejemplo que viene con esta, para realizar el proceso de pruebas. Una característica importante de esta aplicación es que incluye tres niveles de verbosidad. Uno de estos niveles permite ver las características KLT seleccionadas, lo cual confirmaría la validez de la prueba durante la ejecución.

**Objetivo:**

Verificar la efectividad de HandVu para la detección y seguimiento de las manos del usuario.

**Métrica:**

Número de veces que la mano del usuario es detectada. en comparación al número de intentos realizados.

**Grupo Objetivo:**

usuario1: Un usuario con piel oscura.

usuario2: Un usuario de relativa menor edad con piel morena.

usuario3: usuario con piel clara.

usuario4: usuario con piel morena.

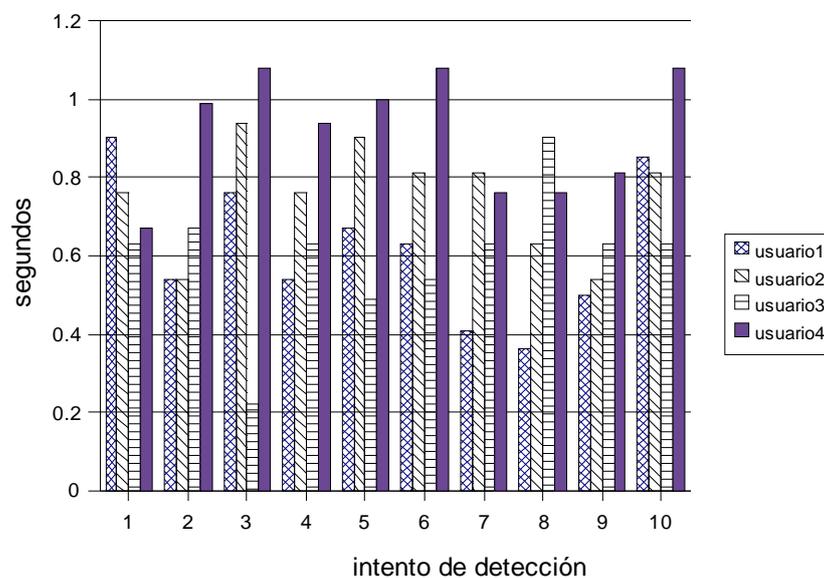


**Gráfico 5.2.** usuarios de Prueba

### Procedimiento:

Para esta prueba, se va a apuntar la cámara hacia una área determinada y se pedirá a los usuarios colocar su mano en el área para medir si HandVu detecta o no la mano del usuario.

### Resultados:



**Gráfico 5.3.** Tiempo de detección de las manos de los usuarios.

### Análisis:

Como se puede observar en el gráfico 5.3, de los resultados obtenidos se puede verificar que se obtuvo un promedio de 0.72 segundos para la detección de la mano del usuario. Esto constituye un argumento adicional a los presentados anteriormente en el capítulo de análisis sobre los cuales se seleccionó a HandVu como herramienta base para la implementación del MDS. El único inconveniente de la herramienta evaluada es que para realizar el inicio de la

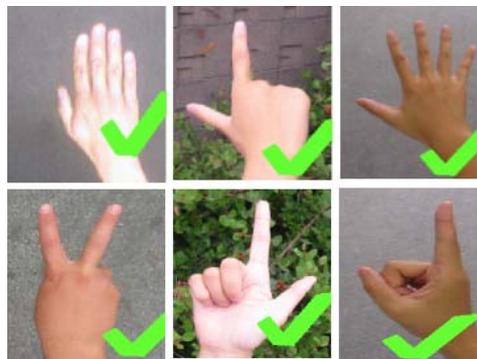
interacción con el sistema, el usuario debe colocar su mano por una región preestablecida.

## Prueba 2 (P-2)

### Descripción de la Prueba

Esta prueba fue relacionada a las posturas dactilológicas que HandVu podía reconocer. Según las especificaciones de la librería, esta incluye por defecto seis posturas dactilológicas que pueden ser identificadas. Refiriéndonos a ellas con los nombres dados por los desarrolladores de HandVu, tenemos:

(a) *closed*, (b) *Lback*, (c) *open*, (d) *victory*, (e) *Lpalm*, y (f) *sidepoint*.



**Gráfico 5.4.** Posturas que por defecto son reconocidas por HandVu.

Durante esta prueba, una vez que el sistema haya detectado la mano del usuario, a cada usuario se le solicitará realizar una postura dactilológica determinada. Se medirá el tiempo desde que la solicitud fue realizada hasta que el HandVu reconozca dicha postura. De esta manera se identificará las posturas dactilológicas que HandVu puede reconocer con un mayor grado de efectividad.

**Objetivo:**

Identificar las posturas dactilológicas que HandVu reconozca con mayor grado de precisión.

**Métrica:**

Tiempo de detección de una postura dactilológica determinada.

Número de identificaciones acertadas.

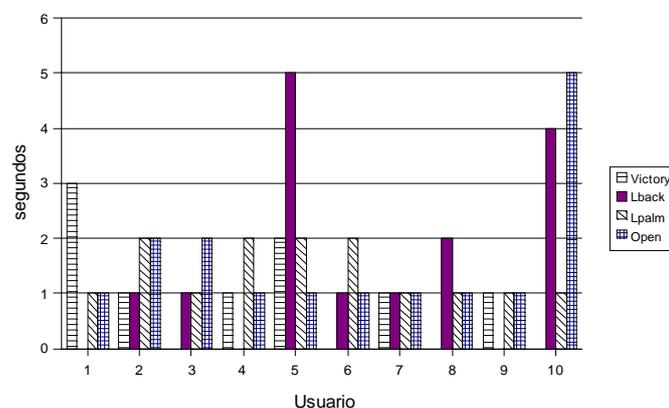
**Grupo Objetivo:**

cinco usuarios constantes de computadoras.

cinco usuarios frecuentes de computadoras.

**Procedimiento:**

Para esta prueba, se va a apuntar la cámara hacia una área controlada, se pedirá a los usuarios realizar una postura determinada. Una vez que el usuario inicie la interacción, se medirá el tiempo que a HandVu le toma reconocer satisfactoriamente dicha postura.

**Resultados:**

**Gráfico 5.5.** Tiempo de detección de las posturas seleccionadas.

Tomando en cuenta los resultados obtenidos, se seleccionaron cuatro de las seis posturas dactilológicas antes mencionadas, para ser usadas en la aplicación de manipulación de imágenes. Para ingresar el comando de traslación, el usuario deberá realizar la postura *victory*; para el comando de rotación, *open*; *Lback* para modificación de tamaño y *Lpalm* para soltar la imagen. Estas posturas fueron seleccionadas ya que fueron reconocidas en un tiempo menor, presentando tiempos promedio de 0.9, 1.6, 1.5 y 1.4 segundos aproximadamente, lo que permite al sistema realizar un reconocimiento en tiempo real, y estas posturas son las que poseen menor margen de error con tasas de 0, 0.1, 0 y 0.1 aproximadamente. En comparación a *closed* y *sidepoint*, posturas que presentaron un elevado número de errores en el momento de realizar el reconocimiento ya que en momentos la postura *sidepoint* era confundida con *lpalm* mientras que *closed* era confundida con *open*, obteniendo una tasa de error de .0.8 y 1 respectivamente en 10 intentos realizados.

### **Prueba 3 (P-3)**

#### **Descripción de la Prueba**

En esta prueba se va a medir la integración del MDS con el Módulo de aplicación a un nivel básico. Para esto se desarrolló la aplicación VisualPad; aplicación que funciona sobre un área de trabajo con un fondo oscuro controlado, por este motivo en esta aplicación el usuario no interactúa directamente sobre las

imágenes sino que las acciones que realice sobre el área controlada son mapeadas sobre la interfase gráfica mostrada en la pantalla. En esta aplicación el usuario puede trasladar imágenes al ingresar la postura *victory* que se empleará para la traslación y *Lback*, la postura que se empleará para soltar la imagen.

**Objetivo:**

Evidenciar el correcto funcionamiento de los módulos en conjunto por medio de la aplicación de prueba visualPad.

**Métrica:**

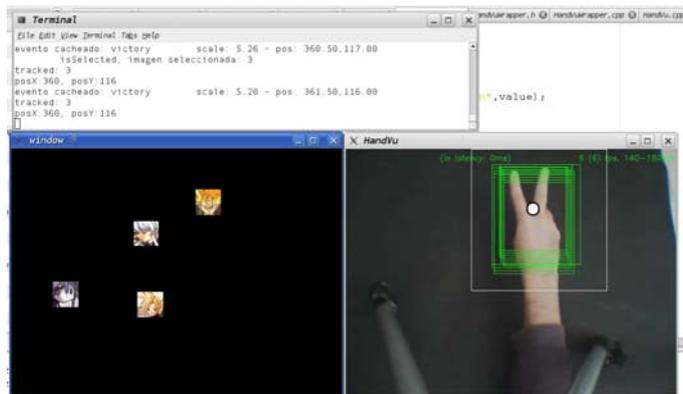
Tiempo de ejecución de la tarea de traslación de imágenes.

**Grupo Objetivo:**

Dos usuarios sin características particulares.

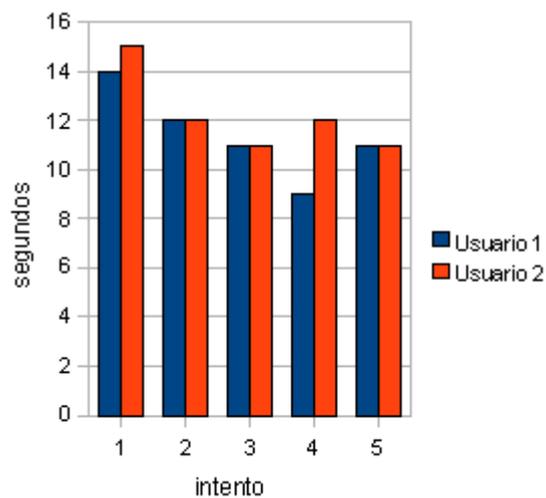
**Procedimiento:**

Para esta prueba, se va a apuntar la cámara hacia un área controlada, se pedirá a los usuarios realizar una tarea determinada relacionada al traslado de alguna imagen mostrada. Se medirá el tiempo que al usuario le toma realizar la tarea planteada.



**Gráfico 5.6.** Aplicación de Pruebas VisualPad.

## Resultados:



**Gráfico 5.7.** Grafico de resultados de pruebas de VisualPad

## Análisis:

Luego de haber realizado esta prueba, se puede verificar que el tiempo que le tomó a los usuarios realizar la tarea planteada fue de aproximadamente 12 segundos. Adicionalmente, se observó que tanto el reconocimiento de las posturas y gestos dactilológicos y el tiempo de respuesta del sistema eran fluidos y rápidos, alrededor de un segundo aproximadamente. Esto muestra que la integración entre el MDS implementado en base a HandVU y visualPad, la aplicación implementada por el MA,

poseen un rendimiento que hace el diseño planteado apto para funcionamiento en tiempo real.

## **Prueba 4 (P-4)**

### **Descripción de la Prueba**

Una vez que el MA ha sido integrado exitosamente con el MDS, se va a probar la efectividad de esta aplicación sobre fondos dinámicos.

Para esto se realizó una prueba similar a P-1 y P-3 con la diferencia que en lugar de que el área de trabajo fuera un fondo oscuro controlado, era un video. Esto brinda un fondo multicolor dinámico que representa un escenario mas complejo que el escenario mas complejo que pueda darse en la aplicación de manipulación de imágenes, en cuanto a diferencia de fondo se refiere. Adicionalmente, para verificar el correcto funcionamiento del sistema empleando otro dispositivo para la implementación del Área de Trabajo, se empleó un televisor CRT.

### **Objetivo:**

Evidenciar la precisión de detección en fondos no controlados.

### **Métrica:**

Tiempo de detección de las manos del usuario

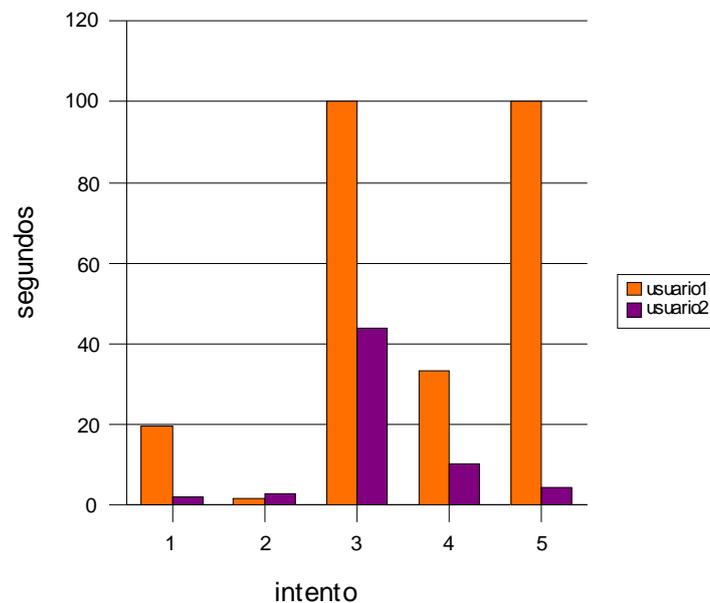
### **Grupo Objetivo:**

Dos usuarios

### Procedimiento:

Para esta prueba, se va a apuntar la cámara hacia un televisor CRT, dicho televisor va a presentar un video con el objetivo de tener un fondo cuyos colores cambien constantemente. Se solicitará a los usuarios colocar su mano sobre el área de trabajo para medir el tiempo que le toma al MDS detectar la mano del usuario.

### Resultados:



**Gráfico 5.8.** Tiempo de detección sobre fondos dinámicos.

### Análisis:

Después de esta prueba se pudo verificar que la efectividad de detección de las manos del usuario se vio afectada, significativamente, por el cambio en el área de trabajo la cual para esta prueba cambiaba constantemente. En algunos casos el tiempo de detección sobrepasaba el minuto, lo cual no era óptimo

para funcionamiento en tiempo real luego de cinco mediciones obteniendo un promedio de 50.87 y 12.63 para el usuario 1 y usuario 2 respectivamente. Ya que en la prueba se empleó una cámara web común, la velocidad de ajuste de imagen de esta sobre un fondo en constante cambio, hace que la detección de las manos del usuario sea igual o mayor a un minuto lo cual hace que bajo estas condiciones el sistema no sea apto para trabajo en tiempo real.

## **Prueba 5 (P-5)**

### **Descripción de la Prueba**

Una vez realizada P-4 y detectado el problema con el ajuste de imagen que suelen tener las cámaras web. Con el objetivo de solucionar este problema se modificó la cámara removiendo el filtro de luz infrarroja y colocando un filtro de luz visible colocando cinta fotográfica en el lente de la cámara. De esta forma la cámara captará luz infrarroja y no luz visible, la cual se encuentra asociada los gráficos mostrados por pantallas LCD, CRT y proyectores emiten luz visible.

### **Objetivo:**

Evidenciar la precisión de detección en fondos no controlados usando una cámara infrarroja.

### **Métrica:**

Tiempo de detección de las manos del usuario

### Grupo Objetivo:

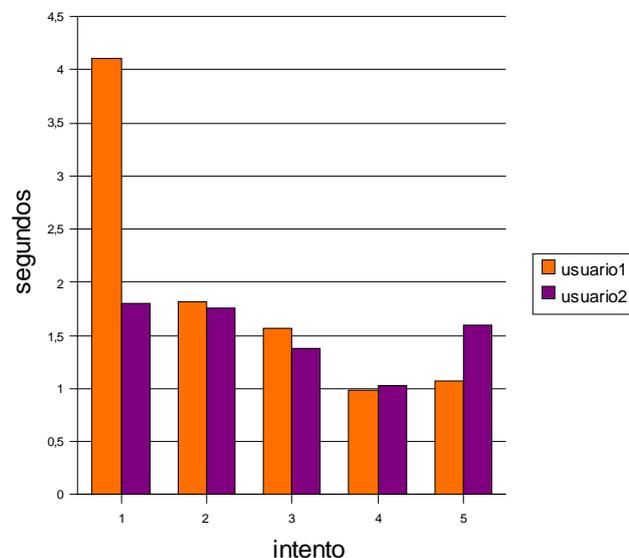
Dos usuarios.

### Procedimiento:

Para esta prueba, se va a apuntar la cámara hacia un televisor CRT, dicho televisor va a presentar un video con el objetivo de tener un fondo cuyos colores cambien constantemente.

Como dispositivo de captura se utilizará una cámara infrarroja. Se solicitará a los usuarios colocar su mano sobre el área de trabajo para medir el tiempo que le toma al MDS detectar la mano del usuario.

### Resultados:

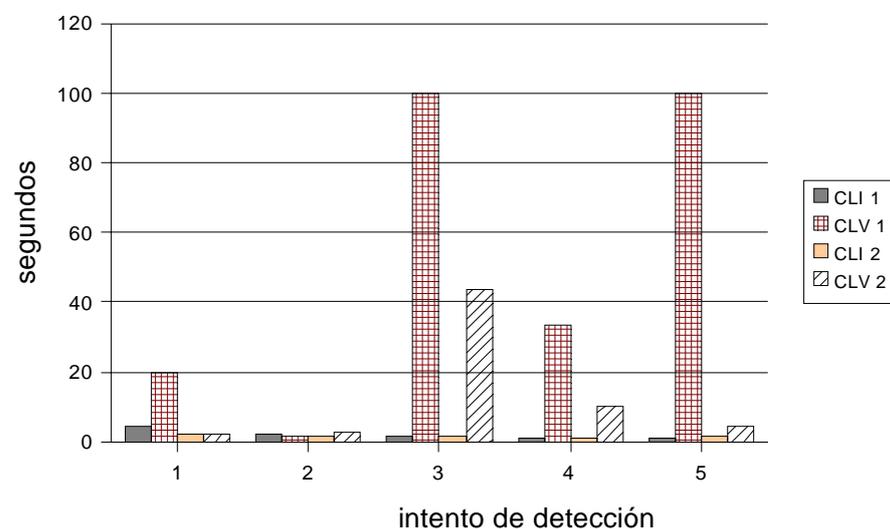


**Gráfico 5.9.** Tiempo de detección de las manos del usuario sobre fondos dinámicos, empleando una cámara infrarroja.

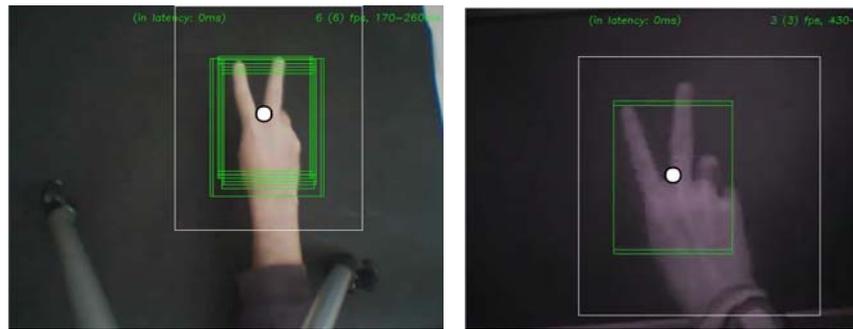
### Análisis:

Luego de realizar esta prueba se pudo constatar cómo al capturar imágenes sobre áreas de trabajo, empleando una cámara

infrarroja los colores presentados en estas pantallas van a ser percibidos como una región oscura. Esto colocó a la prueba en condiciones similares a P-3 en lo referente a fondo controlado, como se puede ver en el gráfico 5.11. Esto hizo que los tiempos de detección del sistema disminuyeran, significativamente, en comparación a los tiempos obtenidos en P-4. Los tiempos promedio de reconocimiento alcanzados fueron 1.91 y 1.51 segundos para el usuario1 y usuario2 respectivamente. En el gráfico 5.10 se comparan los tiempos de detección de dos usuarios empleando una cámara de luz visible (CLV) y una cámara de luz infrarroja (CLI) en cinco intentos donde se midió el tiempo de detección de las manos de los usuarios.



**Gráfico 5.10.** Comparación entre los tiempos de detección de las cámaras empleadas.



**Gráfico 5.11.** Comparación entre los tiempos de detección de las cámaras empleadas.

## Prueba 6 (P-6)

### Descripción de la Prueba

Después de haber realizado las pruebas antes mencionadas para mejorar la efectividad del reconocimiento del lenguaje dactilológico del sistema, se va a realizar una sexta prueba en la cual se evaluará la facilidad con la que el usuario puede aprender a utilizar el sistema.

### Objetivo:

Evaluar la facilidad de aprendizaje de la aplicación de pruebas del sistema.

### Métrica:

Tiempo de ejecución de la tarea de escalar una imagen y trasladarla.

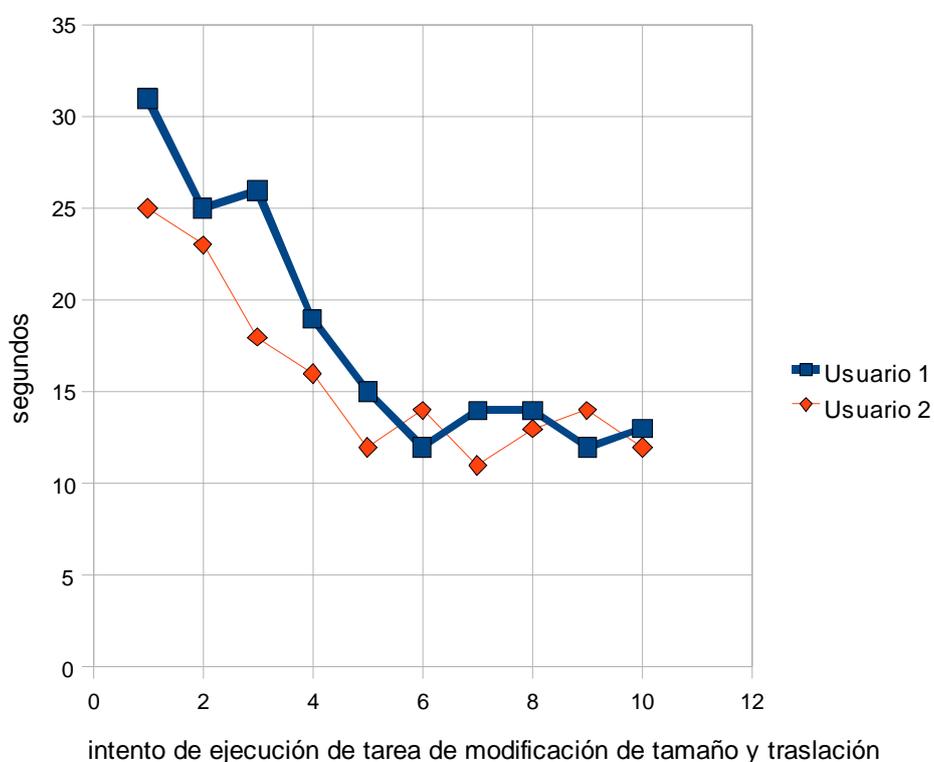
### Grupo Objetivo:

Cualquier usuario.

### Procedimiento:

En esta prueba se levantará el sistema con la aplicación de manipulación de imágenes. A cada uno de los usuarios se les solicitará realizar la tarea escalar una imagen para luego trasladarla y se tomará el tiempo que le tome a cada uno de estos usuarios realizar dicha tarea.

### Resultados:



**Gráfico 5.12** Tiempo que le tomó a cada usuario escalar y trasladar una imagen.

### Análisis:

Como se puede ver en el gráfico 5.12, para la ejecución de la tarea asignada, los tiempos requeridos fueron aproximadamente 13 segundos en promedio. Agregando a esto, como reflejan los tiempos de los siguientes intentos, el tiempo requerido para la

ejecución de cada una de las tareas disminuyó lo cual indica que el número de intentos que le tomó al usuario aprender a utilizar la aplicación de manipulación de imágenes es bajo.

## **Conclusiones**

En este capítulo se presentó las pruebas realizadas durante y después de la implementación del sistema con el objetivo de mejorar su eficacia y su usabilidad. Como se puede ver en los tiempos de detección obtenidos en P-1, la decisión de seleccionar a HandVu para la implementación del MDS, fue correcta. Además por medio de las pruebas utilizando diferentes cámaras web y hardware para la implementación del Área de trabajo como un monitor LCD y el televisor CRT, se puede confirmar la independencia de hardware del diseño del sistema propuesto. Finalmente P-6 muestra el número de intentos que le toma al usuario familiarizarse con el estilo de interacción empleado por el sistema propuesto.

## CONCLUSIONES

- Durante la etapa inicial de este proyecto y durante la revisión de la literatura referente al tema, se pudo constatar diferentes técnicas y herramientas empleadas para el desarrollo de sistemas similares al propuesto. Entre estas herramientas, con el objetivo de realizar un estudio del estado de arte en este campo, se analizó el método planteado por Lee, TouchLib, Bespoke framework y HandVu tomando en cuenta ciertas características requeridas en el sistema. En base a este análisis, se pudo verificar cómo las superficies sensitivas implementadas empleando la tecnología FTIR deben ser empleadas para aplicaciones donde la precisión es un factor requerido en el sistema. Por ejemplo, para aplicaciones en las cuales se realice trazados, prototipazo o simplemente se interactúe con objetos de tamaño cercano al de las yemas de los dedos. Otra ventaja detectada, es la que provee el Wiimote al proporcionar una plataforma físicamente portable, mientras que HandVu proporciona la facilidad de implementar sistemas con una interfaz más natural y compatible con diferentes dispositivos de hardware.

- En lo referente al diseño del sistema, como se presentó en la sección de pruebas, se realizaron implantaciones del sistema empleando diferentes cámaras web para la implementación del Dispositivo de Captura, así como un monitor LCD, un televisor CRT y una pantalla de retroproyección para la implementación del Área de Trabajo. Esto demuestra la independencia de hardware que se propuso en el capítulo de Diseño. Sumando a esto, el uso de HandVu y OpenCV, herramientas que no dependen de un sistema operativo específico, hace que el sistema propuesto cumpla con el requerimiento no funcional de ser portable. Además su arquitectura de capas permite que en caso que haya alguna capa que sea dependiente del sistema operativo en el cual está instalado se pueda realizar implementaciones de esta capa para diferentes sistemas operativos proveyendo una forma de portar el sistema en caso de tener como base librerías o entornos dependientes de la plataforma de software.
- La modificación de la cámara web para funcionar como cámara infrarroja mejoró notablemente la efectividad de reconocimiento del sistema. Además, implícitamente agregó un filtro físico para las imágenes sin afectar el rendimiento del sistema. Al realizar esta modificación, el Dispositivo de Captura del sistema lo hizo apto para poder

funcionar con diferentes tipos de usuario sin verse afectado significativamente por el color de piel de estos.

- Para las implantaciones realizadas todo el hardware empleado se encontraba disponible por lo que el gasto en dispositivos de hardware fue nulo y si se quisiera implementar el costo asociado es muy bajo.
- En lo referente a software, al desarrollar el sistema sobre una distribución de GNU/Linux y empleando herramientas libres y de código abierto, se reducen a cero los costos en esta área.
- Considerando lo antes planteado, y teniendo presente que el diseño del sistema permite el uso de hardware disponible, los costos económicos son reducidos lo máximo posible. En lo referente a carga cognitiva, al emplear HandVu como base sobre la cual se implementó el reconocimiento de posturas dactilológicas, se redujo el conjunto de conocimientos necesarios para el diseño e implementación de algoritmos de visión por computador. Lo cual redujo el problema de reconocimiento al simple uso de una librería existente.
- En lo relacionado a aceptación por parte de los usuarios, como se puede constatar en los resultados de la prueba 6, mostrada en el capítulo de Pruebas, el tiempo que le tomó a los usuarios para realizar las tareas requeridas disminuyó notablemente en los cuatro primeros intentos. Esta

disminución temprana en los tiempos refleja la facilidad con que los usuarios aprendieron la forma de uso del sistema planteado.

- Luego del proceso de pruebas (P-3, P-4 y P-5) se pudo constatar como el método empleado se veía afectado por el color del fondo y como los cambios de iluminación y variaciones de este afectaban el tiempo de detección de la mano del usuario.
- Se obtuvo como producto de esta tesis, un sistema en el cual un usuario puede interactuar sobre una superficie en la cual se presentan imágenes que pueden ser modificadas realizando posturas y gestos dactilológicos sobre las mismas.
- Finalmente, a pesar de su ventaja de tener un bajo costo, el sistema desarrollado posee menor precisión que los sistemas que responden al contacto con las llemas de los dedos. Esto hace que el sistema desarrollado sea útil para aplicaciones como navegación en mapas, navegación en galerías de imágenes y videos, similar a esto, su adaptación a Media Centers como Elisa (7) y Windows Media Center (43) entre otros programas empleados para la administración y reproducción de medios que emplean iconos de tamaño considerable.
- Aplicaciones futuras de interfaces que tomen como base posturas y gestos dactilológicos, no necesariamente sobre

superficies, pueden ser: (a) para el control de dispositivos de salida de video como Televisores, Pantallas de proyección, Monitores, etc. , algo muy similar a lo recientemente presentado por Toshiba (16), Oblong (31) y otras compañías que están por lanzar productos que exploten esta tecnología, (b) para la interpretación de lenguajes de señas de personas sordo-mudas, con el objetivo de eliminar la necesidad de intérpretes, entre otras.

## RECOMENDACIONES

Luego de la implementación del sistema propuesto y de encontrar y corregir errores durante dicho proceso, se puede mencionar lo siguiente:

- El Sol constituye una gran fuente de luz infrarroja, esto hace que el sistema propuesto se vea afectado al ser expuesto directamente a la luz solar o a fuentes intensas de luz infrarroja. Lo que lo hace no apto para funcionar en exteriores durante el día. Por este motivo, se recomienda realizar pruebas de exposición a fuentes intensas de luz infrarroja con el objetivo de determinar las condiciones bajo las cuales fuentes de luz infrarroja afectan el funcionamiento del sistema.
- Durante las implantaciones realizadas en las fases de implementación y pruebas se evaluó el funcionamiento del sistema empleando cámaras web. Para verificar si se producen o no mejoras en la detección y reconocimiento de las instrucciones ingresadas por el usuario, se sugiere la integración y experimentación del sistema con cámaras DV.

- En la implementación realizada, se desarrolló una interfase gráfica de usuario minimalista. Para mejorar la experiencia de uso y la fluidez e inmersividad de las aplicaciones del sistema se recomienda la integración de librerías de sonido y motores de física en el Módulo Gráfico del sistema.

## **REFERENCIAS**

- (1) **A. Agarawala**. Keepin it real: Pushing the desktop metaphor with physics, piles and the pen. Ponencias de la conferencia SIGCHI en Factores Humanos en sistemas computacionales. páginas 1283–1292, 2006.
- (2) **Apple** iPhone. Recuperado en Septiembre 2008.,  
<http://www.apple.com/iphone/>.
- (3) **ARToolkit**. Recuperado en Enero 2008.  
[,http://www.hitl.washington.edu/artoolkit/](http://www.hitl.washington.edu/artoolkit/).
- (4) **Asociación de Maquinaria Computarizada (ACM)**. Recuperado en Mayo 2008. <http://www.acm.org>.
- (5) **Bespoke 3DUI XNA**. Recuperado en Septiembre 2008.  
<http://http://www.bespokesoftware.org/wordpress/index.php>.
- (6) **Centro para el Estudio de Librerías Digitales(CSDL) de la universidad AM de Texas**, Curso de HCI (CPSC 436). Mayo 2008.  
<http://www.csdl.tamu.edu/leggett/courses/436/part1/sld015.htm>.
- (7) **Elisa Media Center**. Recuperado en Junio 2008.  
<http://elisa.fluendo.com/>.
- (8) **C. Engelbart**. Augmenting Human Intellect. Preparado para la Oficina de Investigaciones Científicas de la Fuerza Aérea de EEUU, Octubre 1962.
- (9) **General Public License (GPL)**. Recuperado en Junio 2008.  
<http://www.gnu.org/copyleft/gpl.html>.

- (10) **J. Han.** Low-cost multi-touch sensing through frustrated total internal reflection. Ponencias del XVIII simposio anual ACM en software y tecnologías de interfaces de usuario, páginas 115–118., <http://portal.acm.org/citation.cfm?doid=1095034.1095054>.
- (11) **HandVu.** Recuperado en Septiembre 2007.  
<http://www.movesinstitute.org/kolsch/HandVu/HandVu.html>.
- (12) **Ian Sommerville.** Ingeniería de Software séptima edición. 2004.
- (13) **H. Ishii.** Tangible bits: Towards seamless interfaces, between people, bits and atoms. Ponencias CHI '97, pages 22–27, Marzo 1997.
- (14) **iWindow.** Recuperado en Febrero 2008. <http://www.iwindow.be/>.
- (15) **R. Jota.** Recognizing hand gestures with cali.  
Ponencias EUROGRAPHICS 2006.
- (16) **S. Kei, O. Toshifumi, I. Tsukasa, y F. Masahiro.** Hand gesture interface technology.  
TOSHIBA REVIEW, 63(11), Noviembre 2008.  
<http://www.toshiba.co.jp/tech/review/2008/11/index.htm>.
- (17) **M. Kolsch.** Fast 2d hand tracking with flocks of features and multi-cue integration. Ponencias de la Conferencia en Visión por Computador y Reconocimiento de Patrones 2004, taller (CVPRW'04), página 158, Marzo 2004.
- (18) **M. Krueger.** Videoplance - a virtual reality. Buletín SIGCHI, 16(7):35–40, Enero 1985.

- (19) **M. Krueger**. Videoplace and the interface of the future. The Art of Human Computer Interface Design., páginas 417–422, 1991.
- (20) **J. Lee**. Proyectos empleando el control remoto del Nintendo Wii. Mayo 2008.  
<http://www.cs.cmu.edu/johnny/projects/wii/>.
- (21) **J. Licklider**. Man-Computer Symbiosis. Human Factors in Electronics, IRE Transactions on, HFE-1:4–11, Marzo 1969.
- (22) **C. Lien**. Model-based articulated hand motion tracking for gesture recognition. Image and Vision Computing, 16(2):121–134, Febrero 1998.
- (23) **S. Malik**. Visual touchpad: a two-handed gestural input device. Ponencias de la VI Conferencia Internacional en Interfaces Multimodales, páginas 289–296, Enero 2004.
- (24) **C. Manresa**. Real time hand tracking and gesture recognition for human-computer interaction. (7), 2000.
- (25) **MERL Diamond Touch**. Recuperado en Febrero 2008.  
<http://www.merl.com/projects/DiamondTouch/>.
- (26) **Microsoft Public License (MPL)**. Recuperado en Junio 2008.  
<http://www.microsoft.com/opensource/licenses.msp>.
- (27) **Microsoft Surface**. Recuperado en Mayo 2008.  
<http://www.microsoft.com/surface/index.html>.
- (28) **T. A. Mysliwiec. Fingermouse**: A freehand computer pointing interface. Ponencias de la Conferencia Internacional en reconocimiento automático de rostros y gestos.1995.

- (29) **T. Naoya**. Tele-graffiti: A camera-projector based remote sketching system with hand-based user interface and automatic session summarization. *Journal Internacional de Visión por Computador*, 53:115–133, Julio 2003.
- (30) **C. Nolker**. Illumination independent recognition of deitic arm postures. Ponencias de las XIV Conferencias Annual de la Sociedad IEEE de Electrónica Industrial.
- (31) **Oblong Industries**. Recuperado en Noviembre 2008.  
<http://oblong.com/>.
- (32) **Perceptive Pixel**. Recuperado en Mayo 2008.  
<http://www.perceptivepixel.com/>.
- (33) **F. Quek**. Inductive learning in hand pose recognition. In FG 96: Ponencias de la II Conferencia Internacional en reconocimiento automático de rostros y gestos, páginas 78, 1996.
- (34) **React-In InviDESK**. Recuperado en Mayo 2008.  
<http://invidesk.reactin.com/>.
- (35) **D. Sturman**. Hands-on interaction with virtual environments. Ponencias del simposio anual ACM en software y tecnologías de interfaces de usuario, Noviembre 1989.
- (36) **I. E. Sutherland**. Sketchpad: A man-machine graphical communication system., 23:329–346, 1963.
- (37) **Tactiva Tactapad**. Recuperado en Febrero 2008,  
<http://tactiva.com/tactapad.html>.
- (38) **Touchlib**. Recuperado en Mayo 2008.,  
<http://www.whitenoiseaudio.com/touchlib/>.

- (39) **UbiqWindow**. Recuperado en Febrero 2008.,  
<http://www.ubiqwindow.jp/>.
- (40) **C. von Hardenberg**. Bare-hand human-computer interaction. Ponencias de taller en Interfaces Perceptuales, 15:1–8, 2001.
- (41) **X. Wang**. Tracking of deformable human hand in real time as continuous input for gesture-based interaction. Ponencias de la XVII Conferencia Internacional en Interfaces de usuario inteligentes, páginas 235–242, 2007.
- (42) **Wellner**. Interacting with paper on the digitaldesk. Edición especial: Computer augmented environments: back to the real world, 36:87–96, Julio 1993.
- (43) **Windows Media Center**. Recuperado en Junio 2008.  
<http://www.microsoft.com/windowsxp/mediacenter/default.mspx>.
- (44) **M. Wu**. Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. UIST 03: Ponencias del XVI simposio anual ACM en software y tecnologías de interfaces de usuario, páginas 193–202, 2003.

# **ANEXOS**

### **Anexo A: Tabla de resultados de P-1**

intento	usuario1	usuario2	usuario3	usuario4
1	0.9	0.76	0.63	0.67
2	0.54	0.54	0.67	0.99
3	0.76	0.94	0.22	1.08
4	0.54	0.76	0.63	0.94
5	0.67	0.9	0.49	1
6	0.63	0.81	0.54	1.08
7	0.41	0.81	0.63	0.76
8	0.36	0.63	0.9	0.76
9	0.5	0.54	0.63	0.81
10	0.85	0.81	0.63	1.08
<b>promedio</b>	0.62	0.75	0.6	0.92
<b>promedio total</b>	0.72			

**Tabla A.1:** Tiempo de detección de las manos del usuario empleando HandVu

### **Anexo B: Tabla de resultados de P-2**

usuario	Postura evaluada					
	Victory	Lback	Lpalm	Open	Closed	Sidepoint
1	3	0	1	1	error	error
2	1	1	2	2	error	3
3	0	1	1	2	error	2
4	1	0	2	1	error	error
5	2	5	2	1	error	error
6	0	1	2	1	error	error
7	1	1	1	1	error	error
8	0	2	1	1	error	error
9	1	0	1	1	error	error
10	0	4	1	5	error	error
<b>promedio</b>	0.9	1.5	1.4	1.6	-	-

**Tabla B.1:** Tiempo de reconociendo de las posturas detectadas por HandVu en una prueba realizada a diez usuarios.

**Nota:** En la tabla A.2 se presentan los tiempos de reconocimiento de las posturas dactilológicas evaluadas. La palabra “error” representa a los casos en los cuales HandVu realizó un error en el reconocimiento.

### **Anexo C: Tabla de resultados de P-3**

intento	Usuario 1	Usuario 2
1	14	15
2	12	12
3	11	11
4	9	12
5	11	11
promedio	11.4	12.2

**Tabla C.1:** Tiempo que le tomó los usuarios realizar la tarea de desplazamiento de imágenes empleando el prototipo de pruebas VisualPad.

### **Anexo D: Tabla de resultados de P-4**

intento	usuario1		usuario2	
	CLI 1	CLV 1	CLI 2	CLV 2
1	4.11	19.45	1.8	2.07
2	1.81	1.65	1.75	2.73
3	1.57	100	1.38	43.8
4	0.98	33.23	1.03	10.24
5	1.07	100	1.59	4.31
promedio	1.91	50.87	1.51	12.63

**Tabla D.1:** Comparación de tiempo de detección de las manos de dos usuarios, empleando una cámara de luz visible (CLV) y una cámara de luz infrarroja (CLI) sobre un televisor CRT.

**Anexo E: Tabla de resultados de P-5**

intento	Usuario 1	Usuario 2
1	31	25
2	25	23
3	26	18
4	19	16
5	15	12
6	12	14
7	14	11
8	14	13
9	12	14
10	13	12
promedio	13.33	12.67

**Tabla E.1:** Tiempo que le tomó a dos usuarios realizar la tarea de modificación de tamaño y traslación de imágenes usando el sistema propuesto.