INSTITUTO POLITECNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

No. de Registro: 2011-B007 Serie: Amarilla Noviembre de 2012

Documento Técnico

Sistema de dibujos en 2D multi-touch con reconocimiento de patrones.

Autores:

Fernández Hernández Daniel[1]

Hernández Guerrero Javier Irving[2]

Hernández Hernández Alex[3]

Sánchez Ramírez Gustavo Rafael[4]

RESUMEN

En este reporte se presenta el análisis y la documentación técnica del Trabajo Terminal 2011-B007 titulado “*Sistema de dibujos en 2D multi-touch con reconocimiento de patrones.”*, cuyo objetivo es integrar tecnologías, Kinect™ con una PC para desarrollar un sistema con interfaz de usuario natural con reconocimiento de patrones y para probar la integración se va a desarrollar un sistema de dibujo en 2D tipo paint, donde adicionalmente se cuentan con objetos físicos que representan herramientas para dibujar, además se tienen otros objetos físicos con la función de herramientas básicas para la edición del dibujo.

Palabras clave: *Reconocimiento de Patrones, Sistema multi-touch, Kinect.*

Directores

M. en C. Rafael Norman Saucedo Delgado Dr. José Félix Serrano Talamantes

México D.F. a 22 de Octubre de 2012

|  |  |
| --- | --- |
| 1 [dfh\_ipn\_escom@hotmail.com](mailto:dfh_ipn_escom@hotmail.com) | 3 [alexleche\_14@hotmail.com](mailto:alexleche_14@hotmail.com) |
| 2 [jihdezg03@hotmail.com](mailto:jihdezg03@hotmail.com) | 4 [gsanrmz@gmail.com](mailto:gsanrmz@gmail.com) |

**Índice.**

1. INTRODUCCIÓN. 5

1.1 KINECT™ 5

1.2 OpenNI 5

1.3 OpenCV 6

1.4 Reconocimiento de patrones 6

1.5 TAG’s 7

1.5.1 reacTIVision fiducials 7

1.5.2 Código QR. 8

1.5.3 ARToolkit MARKER 9

1.6 Metodología 10

1.7 Paradigma 10

2. Objetivos 11

2.1 Objetivo General 11

2.2 Objetivos Particulares 11

3. Justificación. 11

4. Estado del arte. 12

4.1 Hand Tracking - Kinect with OpenCV 2.2 & OpenNI 12

4.2 Kinect Active Projection Mapping 13

4.3 Aldebaran Nao Kinect Controller 14

5. Análisis 15

5.1 Módulo 1: Editor básico de dibujo 15

5.1.1 Introducción 15

5.1.2 Objetivo. 15

5.1.3 Análisis y descripción de procesos. 16

5.1.4 Pruebas 18

5.2 Módulo 2: Reconocimiento de trazos a mano alzada 19

5.2.1 Introducción 19

5.2.2 Objetivo 19

5.2.3 Análisis y descripción de procesos 19

5.2.4 Pruebas 20

5.3 Módulo 3: Proyección sobre el área de trabajo 22

5.3.1 Introducción 22

5.3.2 Objetivo 22

5.3.3 Análisis y descripción de procesos 22

5.3.4 Pruebas 23

5.4 Módulo 4: Implementación de herramientas físicas 24

5.4.1 Introducción 24

5.4.2 Objetivo 24

5.4.3 Análisis y descripción de procesos 24

5.4.4 Pruebas 25

5.3 Estudio de Factibilidad. 26

5.3.1 Factibilidad operativa 26

5.3.2 Factibilidad técnica 26

5.3.3 Software actual 27

5.4 Requerimientos. 29

5.5 Especificación de los Requerimientos. 30

6. Desarrollo 41

6.1 Editor de dibujo básico. 41

6.2 Reconocimiento de trazos a mano alzada. 42

6.3 Implementación de herramientas físicas 42

6.3.1 Elección de tag 42

6.3.2 Reconocimiento de Tag’s 43

7 Anexo 47

7.1 Diagrama General de Casos de Uso. 47

7.2 Diagrama General de Clases 60

7.3 Diagrama General de Secuencia 61

7.4 Diagrama General de Estados 62

7.5 Diagrama de Casos de Uso Módulo 1 63

7.6 Diagrama de clases Módulo 1 64

7.7 Diagrama de Secuencia Módulo 1 65

7.8 Diagrama de Estados Módulo 1 66

7.9 Diagrama de Casos de Uso Módulo 2 67

7.10 Diagrama de Clases Módulo 2 68

7.11 Diagrama de Secuencias Módulo 2 69

7.12 Diagrama de Estados Módulo 2 70

7.13 Diagrama de Casos de Uso Módulo 3 71

7.14 Diagrama de Secuencia Módulo 3. 72

7.15 Diagrama de Estados Módulo 3 73

7.16 Diagrama de Casos de Uso Módulo 4. 74

7.17 Diagrama de Clases Módulo 4 75

7.18 Diagrama de Secuencia Módulo 4. 76

7.19 Diagrama de Estados Módulo 4. 77

8. Referencias. 78

# 1. INTRODUCCIÓN.

El presente trabajo muestra el desarrollo de un sistema *multi-touch* para dibujar en 2D tipo *Paint*, donde adicionalmente se cuenta con objetos físicos que representan herramientas para dibujar, además se cuenta con otros objetos como herramientas básicas para la edición del dibujo.

El desarrollo del sistema está enfocado en la integración de tecnologías, en éste caso particular usaremos la herramienta *Kinect™* de Microsoft® conectada a una PC, además el sistema contará con reconocimiento de patrones para la selección de herramientas de dibujo o de edición del mismo.

## 1.1 KINECT™

Kinect™ es un dispositivo que combina una cámara RGB, un sensor de profundidad y un arreglo de micrófonos[1].

Como podemos observar en la Tabla 1, se listan los elementos principales de Kinect junto con su función.

|  |  |
| --- | --- |
| **Elemento** | **Función** |
| Arreglo de micrófonos | Detecta las voces y las aísla del ruido ambiental |
| Proyector de luz infrarroja | Dispara luz infrarroja. |
| Sensor de profundidad cámara infrarroja | Detecta la luz que lanza el proyector de luz infrarroja y genera un canal extra al RGB que trae información de la profundidad de la escena y es similar a un mapa de disparidad estéreo. |
| Motor de inclinación | Ajustar hacía donde están dirigidas las cámaras. |
| Salida de adaptador USB 2.0 | Conectar un cable para conectar vía USB el dispositivo. |
| Cámara RGB | Reconocer los tres colores básicos. Rojo, verde y azul. |

**Tabla 1. Elementos Principales de Kinect™ [2].**

Además cuenta con:

* Triple Core PowerPC 970, 3.2GHz, Hyperthreaded, 2 threads/core.
* 500 MHz ATI graphics card.
* 512 MB RAM.

## 1.2 OpenNI

Se utiliza OpenNI *framework* (Open Source Natural Interaction) versión 1.3.4.6 que es una API desarrollada por *OpenNI organization* que provee una interfaz para dispositivos que ofrecen una interfaz natural para operar o componentes de *middleware*[3], en nuestro sistema lo utilizamos para poder comunicar la PC con Kinect™ y así utilizar de la mejor manera para lo que se diseño Kinect™ que es eliminar los dispositivos físicos para controlar, en este caso, la PC.

Posee una biblioteca de código abierto para poder trabajar con casi todas las capacidades de Kinect™ en Windows, Linux y Mac.

## 1.3 OpenCV

OpenCV (Open Source Computer Vision Library) Es una biblioteca GPL, orientada a la computación visual en tiempo real, utilizada principalmente en el campo de la Interacción Computadora – Humano por sus siglas en inglés HCI (*Human Computer Interaction*) creada y soportada por Intel desde 1999, y tiene amplia documentación.

Actualmente trabajamos con la versión 2.3.1 de la librería OpenCV.

Una de las características más importantes de OpenCV es que las funciones están totalmente optimizadas para los procesadores de arquitectura Intel. Existen versiones para diferentes arquitecturas de procesadores y para diferentes sistemas operativos[4].

## 1.4 Reconocimiento de patrones

Para intentar definir que es el Reconocimiento de Patrones (RP) definiremos algunos conceptos[5].

**Reconocimiento:** Proceso de clasificación de un objeto en una o más clases.

**Objeto:** Es un concepto con el cual representamos los elementos sujetos a estudio. Pueden ser concretos o abstractos.

**Patrón:** Tras los procesos de segmentación, extracción de características y descripción, cada objeto queda representado por una colección (posiblemente ordenada y estructurada) de descriptores.

**Clase:** Es un conjunto de objetos. Al agrupa en clases, se puede hacer de dos formas distintas:

* *Por pertenencias duras*: Un objeto pertenece o no a una clase.
* *Por pertenencias difusas*: Los objetos pertenecen parcialmente a una clase. Existen clases con intersecciones no vacías.

Existen varios intentos para definir al reconocimiento de patrones.

* “La disciplina dedicada a la clasificación de objetos y el pronóstico de fenómenos.”[6]
* “Rama del conocimiento, de carácter multidisciplinario, cuyo objeto de estudio son los procesos de identificación, caracterización, clasificación y reconstrucción sobre conjuntos de objetos o fenómenos, así como el desarrollo de teorías, tecnologías y metodologías relacionadas con dichos procesos.”[6]
* “Es la ciencia que se ocupa de los procesos sobre ingeniería, computación y matemáticas relacionados con objetos físicos y/o abstractos, con el propósito de extraer información que permita establecer propiedades de o entre conjuntos de dichos objetos.”[6]

Con respecto al reconocimiento de patrones, se cuenta con objetos físicos los cuales hacen la función de botones permitiendo eliminar éstos de la interfaz, éstos objetos tienen asignada una imagen binaria que denominamos **tag**, así la cámara de Kinect™ captura imagen del escenario entonces el sistema procesa esa captura donde si se encuentra uno de los objetos físicos se decodifica la imagen y se activa la acción que se tenga asignada a dicha tag. Algunas de las tags tienen más de una acción que se activan con un giro en cierto ángulo esto hace que cambie la función que al principio tenia designada.

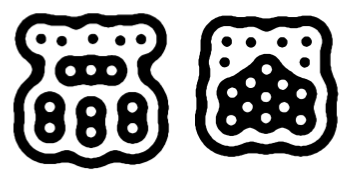
## 1.5 TAG’s

Denominamos tag’s a un conjunto de imágenes binarias en colores blanco y negro que indica el uso de una herramienta.

Analizamos tres tipos de imágenes binarias candidatas, las cuales son: *reacTIVision* *fiducials*, Código QR y *ARToolKit Marker*.

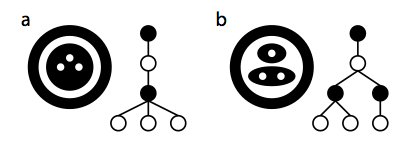
### 1.5.1 reacTIVision fiducials

Son marcadores especiales (*Figura 1*) que se desarrollaron en conjunto con el sistema *reacTIVision* que es un software creado especialmente para el rastreo de éstos marcadores que a grandes rasgos son imágenes binarias, específicamente, en blanco y negro.



**Figura 1. Ejemplos de marcadores para reacTable.**

Para la identificación de cada marcador se basa en una región grafica adyacente y los rectángulos de delimitación. El método combina coincidencias de patrones binarios de gráficos topológicos (*Figura 2*) para el reconocimiento y la identificación con simples técnicas geométricas para calcular la ubicación y orientación de los marcadores[7].



**Figura 2: Algunas simples topologías y su correspondiente gráfico de región adyacente.**

Se utilizan algoritmos genéticos para la identificación de cada marcador, para más detalles consultar[7].

Éstos marcadores están disponibles en PDF para su impresión así como el sistema *reacTIVision* en la página del proyecto[8] y no es necesario producir nuevos marcadores.

### 1.5.2 Código QR.

El código de barras de respuesta rápida por sus siglas en inglés *QR code\** (Quick Response Barcode, *figura 3*) es un sistema que permite almacenar información en un código de barras bidimensional, esto quiere decir que tiene un patrón de arriba hacía abajo, de izquierda a derecha, y puede almacenar alrededor de 7,000 dígitos (véase Tabla 2) mucho más que un código de barras convencional, además con la ayuda de una cámara y un programa especial podemos recuperar la información de cada código. Éste código esta estandarizado ISO/IEC 18004.



**Figura 3. Ejemplo de Código QR**

|  |  |
| --- | --- |
| Numérico | Máximo 7,089 caracteres |
| Alfanumérico | Máximo 4,296 caracteres. |
| Binario | Máximo 2,953 caracteres. |
| Kanji/Kana | Máximo 1,817 caracteres. |

**Tabla 2. Capacidad de datos del código QR.**

Existen versiones del código QR desde la 1 hasta la 40 y cada una tiene diferentes números de módulos (módulo se refiere a los puntos blancos y negros que conforman el código QR)[9].

\* *QR code* es una marca registrada por DENSO WAVE INCORPORATED

Tiene la capacidad de corrección de errores (véase Tabla 3), si una parte del código está dañada, manchada o doblada puede ser interpretado de igual forma.

|  |  |
| --- | --- |
| QR Code Error Correction Capability | |
| Level L | Approx.7% |
| Level M | Approx. 15% |
| Level Q | Approx. 25% |
| Level H | Approx. 30% |

**Tabla 3. Capacidad de corrección de errores Código QR[9].**

La decodificación del código QR puede seguir varios algoritmos a continuación se describe un algoritmo general que puede utilizarse para algunos código de barras en 2D.

1. Binarización de la imagen.

Método de Otsu[10].

2. Corrección de la inclinación.

3. Corrección  geométrica  de la imagen.

4. Obtención de los cuatro vértices de la imagen.

5. Obtención de los nuevos valores de los vértices.

6. Obtener el valor en cada nuevo pixel.

7. Normalización de la imagen.

Existen distintos sistemas que ofrecen la creación de códigos QR así como la decodificación del mismo como ZXing[11].

### 1.5.3 ARToolkit MARKER

Son plantillas de forma cuadrada, que se componen de cuadrado negro con un cuadrado blanco cuatro veces más pequeño que su centro y un dibujo sencillo en el interior del cuadrado blanco, como se muestra en la *figura 4.*



**Figura 4. Ejemplo de ARToolKit Marker.**

Para identificación de la platilla está basada en la detección de las esquinas con ayuda del algoritmo de *fast pose estimation*[12]. Los pasos para el tratamiento de estas plantillas son los siguientes:

1.-La imagen capturada se transforma a una imagen binaria.

2.-Identificamos el marco de color negro.

3.-Extraemos los patrones del dibujo que se encuentra en el interior del marco negro.

4.-Almacenamos los patrones.

5.-Repetimos los primeros tres pasos.

6.-Comparamos los patrones extraídos con los almacenados.

7.-Aplicamos funcionalidad de la imagen.

## 1.6 Metodología

El desarrollo del proyecto se realizará aplicando el modelo incremental. El uso de dicha metodología es porque, presenta la ventaja de ser dinámica y flexible. Permite usar las salidas de las etapas precedentes, como entradas en las etapas sucesivas, y facilita corregir cualquier error detectado o llevar a cabo mejoras en los distintos productos que se generan a lo largo de su aplicación[13].

Esta metodología, así como otras, se basan en la metodología en cascada. El uso de la metodología proporciona:

* Definir actividades a llevarse a cabo en el tiempo de realización del Trabajo Terminal.
* Unificar criterios en la organización para el desarrollo del proyecto.
* Proporcionar puntos de control y revisión.
* Permite seguir secuencias ascendentes o descendentes en las etapas del desarrollo.
* Permite cumplir etapas o fases en paralelo, por lo que es más flexible que la estructurada.

## 1.7 Paradigma

El paradigma será Orientado a Objetos, porque la API usada de OpenCV está en lenguaje C++, que permite la manipulación de objetos, ya que primero definen objetos, para luego enviarles mensajes solicitándoles que realicen sus métodos por sí mismos.

El uso del paradigma proporciona:

* No modela la realidad, sino la forma en que las personas comprenden y procesan la realidad
* Es un proceso ascendente basado en una abstracción de clases en aumento
* Se basa en identificación de objetos, definición y organización de librerías de clases, y creación de macros para aplicaciones específicas
* Utiliza menor cantidad de código
* Es reutilizable

# 2. Objetivos

## 2.1 Objetivo General

Desarrollar un sistema de dibujo en 2D multi-touch con interfaz de usuario natural con reconocimiento de patrones con la herramienta de entretenimiento Kinect™.

## 2.2 Objetivos Particulares

* Documentar el estado del arte.
* Realizar manual de instalación y configuración para usar Kinect™ en la Pc con sistema operativo Ubuntu.
* Dirigir la información recopilada hacia estudiantes con interés en utilizar Kinect.

# 3. Justificación.

Una de las principales características sobre la dificultad del desarrollo de los sistemas *multi-touch*basado en tecnología “reciente” en el caso de Kinect™ era la falta de documentación fiable al momento de plantear este proyecto (Octubre - Diciembre 2011) ya que no se contaba con drivers capaces de explotar todas las características de Kinect™ ni un entorno de desarrollo estable por parte de MS o de la comunidad de software libre. El desarrollo de éste sistema busca contribuir a la creación de documentación formal que permitirá que futuras generaciones tengan mayor cantidad de fuentes fiables y por lo tanto se interesen por la creación de sistemas basados en movimientos, logrando ser un aportador más al crecimiento de dichos entornos.

Además, permitir que los alumnos de la Escuela Superior de Cómputo que se encuentren cursando o tengan interés en el reconocimiento de patrones o semejantes, trabajen con los actuales dispositivos de captura de imagen, siendo en nuestro caso Kinect™ de Microsoft®, dejando a un lado su complejidad y crear un mayor interés, buscando cambiar el enfoque de dicha herramienta en donde el alumno no la vea como un proyecto de Trabajo Terminal sino como prácticas semestrales, lo que brindará mayor competitividad e integración de nuevas tecnologías.

Esta integración de tecnologías ofrece una alternativa al mouse y al teclado pudiendo así evitar enfermedades ya conocidas causadas por éstos dispositivos como lo es el síndrome de túnel carpiano[14].

# 4. Estado del arte.

Actualmente no se cuenta con un sistema de dibujo como el que se pretende realizar, a la fecha de la documentación del estado del arte (Marzo 2012) existen otros trabajos que también manejan Kinect™, OpenCV y OpenNI, elementos con los que se llevará acabo el desarrollo de nuestro sistema , de los cuales vamos a mencionar algunos acontinuación.

Gracias a la aparición de drivers que permiten la interaccion entre el dispositivo Kinect™, que primeramente era exclusivo para la consola de videojuegos Xbox 360 de Microsoft®, y la computadora, se comenzaron a realizar aplicaciones que permiten al usuario tener una interacción más natural, permitiendo ser ellos mismos el control de la aplicación. Debído a que era una tecnología “reciente” existía poca documentación formal acerca de proyectos relacionados.

## 4.1 Hand Tracking - Kinect with OpenCV 2.2 & OpenNI

Es una aplicación sencilla que en primera instancia reconoce la mano de un usuario como un punto permitiendo realizar trazos a mano alzada (figura 5), la aplicación puede cambiar el punto con que se realizar el trazo entre una mano y otra juntando las manos para volverlas a separar así queda realizado el cambio. Esta aplicación también permite la identificación del cuerpo (figura 6) con lo que se toman a ambas manos como puntos de interés, uno de ellos se encarga de realizar el trazo y con el otro se puede seleccionar el color.

La relación que existe entre este trabajo y el que se pensó realizar es que ambos deben poder generar dibujos identificando un punto de interes con el cual se van a hacer los trazos además de seleccionar el color y agregar otra funcionalidades. Éste sistema no cuenta con una documentación y lo único que se puede obtener es lo que se visualiza en un video subido a la red[15].

Figura 5. Hand Tracking

****

Figura 6. Reconocimiento del cuerpo.

## 4.2 Kinect Active Projection Mapping

Es una aplicación que trabaja con Kinect, OpenCV y OpenNI, además comparte la idea de mantener un área de trabajo y una proyeción, de tal modo el usuario interactúa directamente sobre el área designada (*figura 7*).

En este proyecto se utiliza el kinect para reconocer el cuerpo, la posicion de las manos principalmente. El sistema crea un efecto visual en sobre las manos y entre ellas por medio de una imagen que es proyectada sobre una pantalla detras de el usuario[16].

****Esta aplicación ha sido desarrollada en el Computer Fusion Laboratory como parte del programa de ingenieria de la Temple University. La página donde se dan más detalles del proyecto se encuentra aún en construcción. Y por el momento los recursos no están disponibles.

Figura 7. Proyección de imágen sobre un área de trabajo.

## 4.3 Aldebaran Nao Kinect Controller

Es un proyecto donde se controla por medio de Kinect™ a un robot (*figura 8*), organismo autonómo programable y de mediana estatura desarrollado por la empresa Francesa Aldebaran Robotics. Esta aplicación ha sido desarrollada en Technical University Bergakademie Freiberg en alemania por Erik Berger y Heni Ben Amor. Existe información adicional de este proyecto en la página oficial de la universidad[17], Pero se encuentra en idioma Aleman.

Este proyecto no tiene mucha relación en cuanto a la funcionalidad del trabajo que se desea realizar, pero se considera por el hecho de también emplear los elementos que utilizaremos en nuestro sistema.

****

Figura 8. Interacción con robot automáta programable

# 5. Análisis

En el presente apartado se describen los procesos que se llevan a cabo en el “TT 2011-B007”, enfocando en un “Sistema de Dibujo en 2D Multi-Touch con Reconocimiento de Patrones”, el cual está conformado por cuatro módulos:

* Editor Básico de Dibujo
* Reconocimiento de trazos a mano alzada
* Proyección sobre el área de trabajo
* Implementación de las herramientas físicas para el dibujo

Se mostrará el análisis de los cuatro módulos ya antes mencionados. Con el análisis correspondiente a los módulos tres y cuatro se fijarán las especificaciones necesarias, además se mencionarán todas aquellas problemáticas detectadas en los procesos.

Se mostrará el estudio de factibilidad correspondiente al proyecto, posteriormente se expondrá el documento de definición de requerimientos, para finalizar con la especificación de requerimientos.

En el anexo se mostrarán los diagramas generales y de los primeros dos módulos que representan los procesos mencionados.

## 5.1 Módulo 1: Editor básico de dibujo

### 5.1.1 Introducción

En las sub-secciones siguientes, se definen los procesos que se llevan a cabo en el modulo uno, el cual se estructura de funciones básicas de dibujo, las cuales son: mano alzada, línea, circunferencia, elipse y polígonos (3 a 6 lados), así como detectar los problemas que puedan surgir.

### 5.1.2 Objetivo.

Analizar los procesos necesarios para el desarrollo de un editor de dibujo y detectar posibles problemas por medio de la realización de pruebas que se documentan para precisar los requerimientos y redactar una propuesta con bases firmes y confiables.

### 5.1.3 Análisis y descripción de procesos.

A continuación se describen los procesos correspondientes al módulo uno.

#### 5.1.3.1 Trazo a Mano Alzada

**Objetivo**: Realizar un trazo a mano alzada, es decir a pulso.

**Descripción**: Es un trazo que no requiere de reglas ni herramientas de medición exactas o auxiliares, solo con el movimiento de tu muñeca o pulso, Pn son el conjunto de puntos que conforman el trazo. Dado el conjunto de puntos Pn(xn,yn) se encenderá un pixel por cada punto.

**Datos de entrada:** Pn(xn,yn)

**Datos de salida:** Trazo a mano alzada

**Problemas:** la velocidad de desplazamiento al aumentar reduce el conjunto de puntos

#### 5.1.3.2 Trazar una Línea recta

**Objetivo:** Realizar trazos rectos a partir de dos puntos.

**Descripción:** Es la sucesión continua de puntos en una misma dirección, donde P1(x1,y1) es la posición de partida y P2(x2,y2) es la posición final. Dados los puntos P1(x1,y1) y P2(x2,y2) se traza una línea entre estos por medio de la siguiente ecuación.

**Datos de entrada:** P1(x1,y1) y P2(x2,y2)

**Datos de Salida:** Línea recta entre P1(x1,y1) y P2(x2,y2)

**Problemas**: Se puede apreciar una pequeña deformación en la línea, al no poder segmentar un pixel.

#### 5.1.3.3 Trazar una Circunferencia

**Objetivo:** Dibujar una superficie plana limitada por una circunferencia.

**Descripcion:** Es una curva cerrada y plana en la que todos sus puntos están a la misma distancia, de otro punto fijo, que llamamos centro. Dados los puntos P1(x1,y1) y P2(x2,y2) se traza un lugar geométrico de los puntos de un plano que equidistan de otro punto fijo y coplanario llamado centro en una cantidad constante llamado radio.

**Datos de entrada:** P1(x1,y1) y P2(x2,y2)

**Datos de salida:** Circunferencia entre P1(x1,y1) y P2(x2,y2)

**Problemas:** Se puede apreciar una deformación mayo cuando la distancia entre los puntos sea menor, al ser pixeles de forma cuadrada.

#### 5.1.3.4 Trazar un Polígono

**Objetivo:** Dibujar una figura plana compuesta por una secuencia de segmentos rectos.

**Descripción:** Es una figura plana compuesta por una secuencia finita de segmentos rectos consecutivos que cierran una región en el espacio. Dados dos puntos P1(x1,y1) y P2(x2,y2), se genera una circunferencia la cual se divide entre el numero de lados obteniendo así intersecciones llamadas vértices, las cuales se unen atreves de segmentos de línea recta.

**Datos de entrada:** P1(x1,y1) y P2(x2,y2)

**Datos de salida:** Polígono entre P1(x1,y1) y P2(x2,y2)

**Problemas:** Se puede apreciar una deformación, cuando se le da una dimensión, al no poder segmentar un pixel

#### 5.1.3.5 Trazar una elipse

**Objetivo**: Es una circunferencia aplastada, una curva simetrica cerrada. Dibujar una curva plana y cerrada, simétrica respecto a dos ejes perpendiculares entre sí.

**Descripción**: dados los puntos P1(x1,y1) y P2(x2,y2) se traza una curva plana cerrada que es simétrica respecto a dos ejes, los cuales constan de focos (puntos fijos), eje focal (recta que pasa por los focos), centro (punto de intersección de los focos) y de los ejes con la siguiente ecuación.

**Datos de entrada:** P1(x1,y1) y P2(x2,y2)

**Datos de salida:** Elipse entre P1(x1,y1) y P2(x2,y2)

**Problemas:** Se puede apreciar una deformación mayor cuando la distancia entre los puntos sea menor, al ser pixeles de forma cuadrada.

### 5.1.4 Pruebas

En el presente apartado se muestran las pruebas que se llevaron a cabo en el Modulo 1 para determinar si cumple con todos los requerimientos y condiciones de satisfacción, para verificar si la funcionalidad es la correcta, se uso un plan de pruebas del tipo caja negra, en la cual se tiene un conjunto de entradas y se verificaba si la salida es la correcta, ya que el proceso interno de las funciones no es complejo se descarto la posibilidad de evaluar con un plan de pruebas del tipo caja blanca. En la Tabla 1 se muestra los resultados de las pruebas que se llevaron a cabo para el modulo 1 y las correcciones para cumplir con las condiciones de satisfacción y requerimientos:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. de prueba | Requerimientos probados | Entrada | Resultados | Correcciones |
| 1 | R1:Trazo Mano Alzada | Posición del puntero | Al aumentar la velocidad del desplazamiento del mouse, no traza en algunos intervalos | Cambiar la forma en que se obtienen las coordenadas del puntero |
| 2 | R2:Trazar Líneas Rectas  R3:Trazar líneas rectas en varias direcciones | Posición del puntero al presionar y Posición del puntero al soltar | Trazo correcto | Ninguna |
| 3 | R4:Trazar Circunferencia | Posición del puntero al presionar y Posición del puntero al soltar | No traza circunferencia si el trazo empieza de abajo hacia arriba | Validar para cualquier posición |
| 4 | R5:Trazar Elipse | Posición del puntero al presionar y Posición del puntero al soltar | Elipse incompleta | Modificar los arcos a 180 grados de cada eje |
| 5 | R6:Trazar Polígonos  R7:Tipo de polígono de 3-6 lados | Posición del puntero al presionar y Posición del puntero al soltar | Al trazar polígonos de 5-6 lados, la figura se deforma si es trazo es de abajo hacia arriba | Validar para cualquier posición y precisar un poco más los vértices |

**Tabla 4. Resultados de las pruebas obtenidas en el Modulo 1**

## 5.2 Módulo 2: Reconocimiento de trazos a mano alzada

### 5.2.1 Introducción

En las sub-secciones de este apartado, se definen los procesos que se llevan a cabo en el modulo dos, donde se reconoce el trazo a mano alzada por medio de kinect, dicho trazo será realizado por el usuario en el área de trabajo que estará delimitada, así como detectar los problemas que se puedan manifestar en los procesos.

### 5.2.2 Objetivo

Analizar los problemas que surjan en los procesos que involucra la realización de éste módulo, de igual manera se documentarán las pruebas hechas con esto corregiremos errores para cumplir con los requerimientos establecidos.

### 5.2.3 Análisis y descripción de procesos

Este apartado describe los procesos pertinentes para la implementación del módulo dos.

#### 5.2.3.1 Integración del Kinect

**Objetivo:** la computadora debe reconocer el dispositivo kinect

**Descripción**: Mediante el driver “OpenNI” se hará la comunicación entre el dispositivo y la PC para poder realizar capturas de imágenes, utilizar y procesar la información recibida.

**Datos de entrada:** Datos recibidos por el sensor

**Datos de salida:** Detección exitosa del kinect

**Problemas:** No poder reconocer el dispositivo

#### 5.2.3.2 Detección de Imagen

**Objetivo:** Detectar la imagen infrarroja mediante el dispositivo kinect

**Descripción:** Se capturarán los datos de profundidad de la escena por medio de la camara infrarroja.

**Datos de entrada:** Datos de profundidad.

**Datos de salida:** Imagen de la escena con profundidad.

**Problemas:** Si la imagen se sale del rango para el reconocimiento no se podrá detectar.

#### 5.2.3.3 Reconocimiento del desplazamiento

**Objetivo:** Detectar que un punto de interés realiza un desplazamiento.

**Descripción**: Mediante el kinect se captura como se desplaza el punto de interés siguiendolo con un marcador que enfoque esa área.

**Datos de entrada:** P1(x1,y1)

**Datos de salida:** Pn(xn,yn)

**Problemas:** si la velocidad de desplazamiento aumenta drásticamente se reduce la captura de los puntos

#### 5.2.3.4 Dibujar a mano alzada con el dedo

**Objetivo:** Poder plasmar el movimiento del área de interés en un trazo.

**Descripción:** Se levanta la mano y se inicia un movimiento libre dentro del rango de visión de kinect, procesándolos y reflejándolos en el encendido de los pixeles del trazo sobre el editor.

**Datos de entrada:** P1(x1,y1)

**Datos de salida:** Pn(xn,yn), trazo realizado

**Problemas:** Si la velocidad de desplazamiento varía de la velocidad de procesamiento, el trazo realizado podría no ser igual al movimiento del dedo.

### 5.2.4 Pruebas

En el presente apartado se muestran las pruebas que se llevaron a cabo en el módulo dos para determinar la funcionalidad y el cumplimiento de los requerimientos, para verificar si la funcionalidad es la correcta, se uso un plan de pruebas de tipo caja blanca, ya que el procesamiento dentro de las funciones es de suma importancia, en ellas encontraremos envíos de paquetes de datos, reconocimiento de la extremidad y tiempo de comunicación entre el modulo 1 y modulo2. En la Tabla 5 se muestra los resultados obtenidos al revisar con detenimiento cada función del modulo dos y la comunicación con el modulo uno.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. de Prueba | Requerimientos Probados | Prueba | Resultados | Correcciones |
| 1 | R8:Integracion de kinect | Verificar la comunicación de kinect a PC | El dispositivo kinect y la PC no se comunican entre si | Verificar si el driver y sus respectivos paquetes están instalados en las rutas correctas |
| 2 | R9:Detección del dedo índice | Verificar si al colocar la mano dentro del área de visión del kinect se identifica el dedo índice | El algoritmo no funciona correctamente únicamente se obtiene la palma de la mano como punto de referencia | Modificar el cálculo al segmentar la imagen y verificar si el contorno de la mano está completamente cubierto |
| 3 | R10:Renococer Desplazamiento del dedo índice | Verificar la posición del dedo índice al desplazarse dentro de la visión del kinect | Únicamente se tiene el seguimiento de la palma. | Corregir el R9, ya que se depende del mismo |
| 4 | R11:Dibujar a mano alzada con el dedo índice | Desplazar el dedo índice dentro de la visión del kinect simulando un trazo a mano alzada | Se dibujo la trayectoria, sin embargo el seguimiento fue de la palma | Corregir el R9 |
| 5 | R12:Dibujar dentro del módulo 1 | Proyectar el trazo obtenido en el R11 dentro del modulo 1 | Error en la Conexión | Verificar el envió de paquetes |

**Tabla 5. Pruebas realizadas al modulo 2**

## 5.3 Módulo 3: Proyección sobre el área de trabajo

### 5.3.1 Introducción

En las sub-secciones del modulo 3, se definen los procesos que se llevan a cabo en el modulo, el cual se refiere a la imagen que nos proporcionará un proyector en el área de trabajo, donde el usuario podrá visualizar los trazos realizados. Así mismo se pretende detectar problemas en los procesos, con ello se realizo una tabla de pruebas, que se describe en la sub-sección 5.3.4.

### 5.3.2 Objetivo

Analizar los procesos pertinentes del módulo 3 por medio de distintas configuraciones tanto del proyector como de Kinect para el desarrollo del mismo cumpliendo completamente con el requerimiento establecido.

### 5.3.3 Análisis y descripción de procesos

Este apartado describe los procesos correspondientes al módulo dos.

#### 5.3.3.1 Mostrar trazos realizados

**Objetivo:** Poder vizualizar en el área de trabajo los trazos realizados.

**Descripción:** Se capturan los movimientos del dedo en una secuencia de video continuo procesándolos y reflejándolos en el encendido de los pixeles de acuerdo al trazo sobre el editor.

**Datos de entrada:** Secuencia de video continuo

**Datos de salida:** Pn(xn,yn), trazo realizado

**Problemas:** Si la velocidad de desplazamiento varía de la velocidad de procesamiento, el trazo realizado podría no ser igual al movimiento del dedo.

### 5.3.4 Pruebas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. prueba | Requerimientos probados | Superficie | Resultados | Correcciones |
| 1 | R13: Trabajar conjuntamente proyector y *Kinect* | Vidrio | Se puede trabajar conjuntamente sin problema alguno. | Ninguna |
| 2 | R14: Proyectar sobre el área de trabajo | Vidrio | La imagen de salida traspasa el vidrio | Cambiar la superficie |
| 3 | R14: Proyectar sobre el área de trabajo | Vidrio polarizado | La captura de la cámara se reduce | Cambiar superficie |
| 4 | R15: Trabajar sin interferencia de la sombra que produzca la mano | Vidrio  Polarizado | Se puede trabajar pero se reduce la velocidad de captura de la tag por la condición de luz | Cambiar superficie. |
| 5 | R14: Proyectar sobre el área de trabajo | Pared | Imagen proyectada, no es la más apta, solución temporal | Identificar otro tipo de superficie adecuado y de bajo costo |
| 6 | R15: Trabajar sin interferencia de la sombra que produzca la mano | Pared | No afecta la sombra ya que no se proyecta directamente en el área de trabajo | Identificar otro tipo de superficie adecuado y de bajo costo. |

## 5.4 Módulo 4: Implementación de herramientas físicas

### 5.4.1 Introducción

En las sub-secciones siguientes, se definen los procesos que se llevan a cabo en el modulo 4, en el cual se implementan las tags, con las que el usuario puede realizar un trazo ó hacer cambios en el dibujo, dichas opciones de edición son: cambiar de color, mover de posición y escalar el dibujo. Así como detectar los problemas que surjan en los procesos, con ello se realizo una tabla de pruebas, que se describe en la sub-sección 5.4.4.

### 5.4.2 Objetivo

Analizar los procesos involucrados para el reconocimiento de las tags aplicado lo investigado sobre distintos algoritmos para reconocer la imagen binaria elegida y tener una implementación completa de acuerdo a los requerimientos.

### 5.4.3 Análisis y descripción de procesos

Este apartado describe los procesos correspondientes al desarrollo del módulo cuatro.

#### 5.4.3.1 Selección de herramienta de trabajo

**Objetivo:** Identificar la tag colocada para iniciar un trazo o la edición de un dibujo.

**Descripción:** Se captura imagen de la tag y se procesa, deacuerdo a que tag este colocada se internamente se selcciona la opción correspondiente para iniciar el trazo de un dibujo o la edición del mismo.

**Datos de entrada:** Secuencia de video continuo

**Datos de salida:** No hay datos de salida visibles al usuario, internamente queda seleccionada una opción

**Problemas:** La tag puede no estar posicionada correctamente para su identificación.

### 5.4.4 Pruebas

En el presente apartado se muestran las pruebas que se llevaron a cabo para el Modulo 4, así determinar si cumple con todos los requerimientos y condiciones de satisfacción, para verificar si la funcionalidad es la correcta, se uso un plan de pruebas del tipo caja negra, en el cual, se tiene un conjunto de entradas y se verifica si la salida es la correcta, ya que el proceso interno de las funciones no es complejo, se descartó la posibilidad de evaluar con un plan de pruebas del tipo caja blanca. En la Tabla 7 se muestra los resultados de las pruebas que se llevaron a cabo para el módulo y las correcciones para cumplir con lo propuesto:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. de prueba | Requerimientos probados | Entrada | Resultados | Correcciones |
| 1 | R16:Reconocimiento de la herramienta | Tag de alguna herramienta | Las herramientas se reconocieron a excepción de línea y mano alzada | Cambiar el grosor del dibujo interno de la tag |
| 2 | R17:Dibujar con el uso de la herramienta | Colocar la tag frente a la cámara y después realizar desplazamiento de mouse sobre el área de trabajo | Trazo correcto | Ninguna |
| 3 | R18,R19,R20:Cambios de la figura con el uso de la herramienta | Colocar la tag que indica modificación y realizar un click dentro de la figura | No se reconoció las tags de modificación | Modificar dibujo interior de las herramientas |
| 4 | R16:Reconocimiento de la herramienta | Tag Mano alzada | Reconocida | Ninguna |
| 5 | R16:Reconocimiento de la herramienta | Tag Linea | Reconocida | Ninguna |
| 6 | R18,R19,R20:Cambios de la figura con el uso de la herramienta | Colocar la tag que indica modificación y realizar un click dentro de la figura | Reconoció tag pero no identifico giro | Modificar método de reconocimiento de rotación |

## 5.3 Estudio de Factibilidad.

Después de definir la problemática presente y establecer las consideraciones de hardware y software, es conveniente realizar un estudio de factibilidad para el TT-2011-B007 “Sistema de Dibujo en 2D multi-touch con reconocimiento de patrones”, se muestra el análisis técnico y operativo que implica la implementación del sistema.

### 5.3.1 Factibilidad operativa

Para un mejor alcance como proyecto se desarrollará una aplicación lo más amigable posible, de tal forma que sin mucha dificultad, un usuario no experimentado, pueda adaptarse y aprovechar al máximo las facilidades que este brinde.

Los usuarios podrán visualizar la información que ellos soliciten, al interactuar con el editor y las herramientas conjuntamente con el *Kinect*, claro que solo con las operaciones mencionadas, no se podrá hacer algo que no esté indicado.

Al ir implementando módulo por módulo, facilitará el trabajo para el avance de los posteriores módulos, y darse una idea de las posibles *tag’s* que se necesitarán usar para el Trabajo Terminal.

Estamos conscientes de aceptar los cambios y mejoras que el trabajo terminal ofrezca dentro del periodo de realización, llegando a la conclusión de que el sistema es factible operativamente, ya que se cuenta con la aceptación y la tecnología para desarrollar el sistema con éxito.

### 5.3.2 Factibilidad técnica

Actualmente para la realización del proyecto se cuenta con cuatro laptops, un *Kinect* y un proyector, prestado por la Escuela Superior de Computo (ESCOM). Estas se utilizarán para desarrollar y hacer pruebas colectivamente con el *Kinect*, aunque en cuestiones con el dispositivo solo es necesaria una.

De acuerdo a la tecnología necesaria para la implementación del Sistema se evaluo bajo dos enfoques: hardware y software

En cuanto a hardware existente, no se requirió realizar inversión inicial para la adquisición del mismo, ya que con lo que se contaba satisfacía los requerimientos establecidos, tanto para el desarrollo y puesta en funcionamiento del sistema propuesto.

A continuación se muestra la descripción de las laptops y del kinect:

* Una Apple MacBook

|  |  |
| --- | --- |
| Procesador | Intel Core 2 Duo |
| Velocidad del Procesador | 2.26 GHz |
| Numero de Procesadores | 1 |
| Memoria RAm | 2 GB |

* Una Dell Inspiron 1545

|  |  |
| --- | --- |
| Procesador | Pentium Dual-Core |
| Velocidad del Procesador | 2.20 GHz |
| Numero de Procesadores | 1 |
| Memoria RAm | 3 GB |

* Una HP Pavilion dv4

|  |  |
| --- | --- |
| Procesador | AMD Athlon II Dual Core |
| Velocidad del Procesador | 2.00 GHz |
| Numero de Procesadores | 1 |
| Memoria RAm | 3 GB |

* Un Kinect

|  |
| --- |
| Triple Core PowerPC 970, 3.2GHz, Hyperthreaded, 2 threads/core. |
| 500 MHz ATI graphics card. |
| 512 MB RAM |
| Arreglo de micrófonos |
| Proyector de luz infrarroja |
| Sensor de profundidad  cámara infrarroja |
| Motor de inclinación |
| Salidad de adapatador USB |
| Camara RGB |

### 5.3.3 Software actual

En cuanto al software ,no amerita una inversión alguna, ya que el sistema operativo, el driver y la API son versiones libres, que a continuación se mencionan:

|  |  |
| --- | --- |
| Sistema operativo: | *Linux Ubuntu 11.04* y versiones posteriores, |
| Driver: | *OpenNI versión 1.3.4.6* |
| API para visión por computadora: | *OpenCv versión 2.3.1* |

**Ventajas de Linux**

|  |  |  |
| --- | --- | --- |
| Característica | Windows | Linux Ubuntu |
| Familiaridad | Si | Si |
| Seguridad | No | Si |
| Precio | No | Si |
| Drivers | Si | Si |

Tabla 6. Tabla Comparativa de Ventajas entre Windows y Linux

**Comparativa S.O**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sistema Operativo | Driver  OpenNi | Middleware o utilidades | IDE | Lenguaje | API para visión por computadora |
| Windows 7 |  | Nite | Visual Studio 2008 en adelante,  Eclipse,  CodeBlocks | C ++  Phyton | Processing |
| **Ubuntu** |  | **Nite** | **QT,**  **Eclipse,**  **(Otros IDE’s )** | **C ++**  Phyton | **OpenCV** |
| Mac OSX |  | Nite | XCode | C ++  Phyton | OpenCV |

Tabla 7. Comparativa de Sistemas Operativos

En conclusión se utilizará la distribución *Ubuntu* del sistema operativo *Linux*, ya que es un software libre y no requiere de herramientas propietarias; el *Driver OpenNi* que nos proporciona el *middleware Nite*, el cual es soportado por la empresa *PRIMESENSE*, la cual desarrollo el dispositivo *Kinect* y *OpenCv* que es una API para visión por computadora y se utilizado en proyectos con *Kinect.*

## 5.4 Requerimientos.

R1: El módulo 1 del sistema permite dibujar a mano alzada

R2: El módulo 1 del sistema permite dibujar líneas de diferente longitud

R3: El módulo 1 del sistema permite crear líneas en diferente dirección (ubicación)

R4: El módulo 1 del sistema permite dibujar círculos de diferente circunferencia

R5: El módulo 1 del sistema permite dibujar una elipse de distintas dimensiones

R6: El módulo 1 del sistema permite dibujar polígonos de diferentes tamaños

R7: El módulo 1 del sistema permite dibujar polígonos de 3 a 6 lados dependiendo de la tag utilizada.

R8: El módulo 2 del sistema permite la integración con *Kinect*

R9: El módulo 2 del sistema permite la detección del dedo índice

R10: El módulo 2 del sistema permite reconocer el desplazamiento del dedo índice

R11: El módulo 2 del sistema permite dibujar a mano alzada con el dedo índice

R12: El módulo 2 del sistema permite proyectar en el editor básico de dibujo la acción realizada por el dedo índice

R13: El módulo 3 del sistema permite trabajar conjuntamente proyector y *Kinect*

R14: El módulo 3 del sistema permite proyectar sobre el área de trabajo

R15: El módulo 3 del sistema permite trabajar sin interferencia de la sombra que produzca la mano

R16: El módulo 4 del sistema permite reconocer la herramienta *(tag*)

R17: El módulo 4 del sistema permite dibujar con la herramienta (*tag*) de la figura a usar

R18: El módulo 4 del sistema permite realizar la selección de escalar una figura

R19: El módulo 4 del sistema permite realizar la selección de mover la figura

R20: El módulo 4 del sistema permite realizar la selección de cambiar el color de la figura.

## 5.5 Especificación de los Requerimientos.

***R1:*** El módulo 1 del sistema permite dibujar a mano alzada

**Objetivo:** el módulo permite al usuario realizar un trazo a mano alzada, es decir a pulso

**Descripción:** Permite el dibujo a mano alzada, el cual se realiza simplemente con el dedo, u otro instrumento que lo simule, es dibujar a pulso, como se muestra en la *Figura 9*.

**Datos de entrada:** posición inicial y final

**Datos de salida:** Trazo realizado

**Pre-condiciones:** Posición de partida.



Figura 9. Módulo 1 - Dibujar a Mano Alzada

***R2:*** El módulo 1 del sistema permite dibujar líneas de diferente longitud

**Objetivo:** Realizar trazos rectos sueltos y dinámicos

**Descripción:** Permite el dibujo de una sucesión continua de puntos (trazado) con diferentes longitudes, según la desea por el usuario, como se muestra en la *Figura 10*, el cual se realiza simplemente con el dedo, u otro instrumento que lo simule, es dibujar a pulso.

**Datos de entrada:** posición inicial y final

**Datos de salida:** Trazo realizado (línea)

**Pre-condiciones:** Posición de partida.

Figura 10. Módulo 1 - Lineas de Diferente Longitud



***R3:*** El módulo 1 del sistema permite crear líneas en diferente dirección (ubicación)

**Objetivo:** Realizar un trazo recto suelto en una cierta posición.

**Descripción:** Permite el dibujo de una sucesión de puntos (trazado) en cierta posición de coordenadas (X1, Y1) iniciales y (X2, Y2) finales, según donde se sitúe el usuario, como se muestra en la *Figura 11*, el cual se realiza simplemente con el dedo, u otro instrumento que lo simule, es dibujar a pulso.

**Datos de entrada:** posición inicial y final

**Datos de salida:** Trazo realizado (línea)

**Pre-condiciones:** Posición de partida.

Figura 11. Módulo 1 - Líneas en Diferentes Direcciones



***R4:*** El módulo 1 del sistema permite dibujar circunferencias de distintos radios

**Objetivo:** Dibujar una superficie plana limitada por una circunferencia.

**Descripción:** Es el lugar geométrico de los puntos de un plano que equidistan de otro punto fijo y coplanario llamado centro en una cantidad constante llamada radio, el usuario decidirá el tamaño de la circunferencia, como se muestra en la *Figura 12*.

**Datos de entrada:** posición inicial y final

**Datos de salida:** Circunferencia realizada

**Pre-condiciones:** Posición de partida.



Figura 12. Módulo 1 - Circunferencias de Distintos Radios

***R5:*** El módulo 1 del sistema permite dibujar una elipse de distintas dimensiones

**Objetivo:** El módulo del sistema permite dibujar una superficie curva plana simétrica a 2 ejes.

**Descripción:** El módulo del sistema permite el dibujo de una región curva del plano simétrica a 2 ejes (elipse), la cual el usuario establecerá la dimensión con el movimiento de su dedo, como se muestra en la *Figura 13*.

**Datos de entrada:** posición inicial y final

**Datos de salida:** elipse realizada

**Pre-condiciones:** Posición de partida.



Figura 13. Módulo 1 - Elipse de distinta dimensión.

***R6:*** El módulo 1 del sistema permite dibujar polígonos de diferentes tamaños

**Objetivo:** el módulo del sistema permite dibujar una figura plana compuesta por una secuencia de segmentos rectos.

**Descripción:** El módulo del sistema permite al usuario dibujar una figura plana compuesta por lados (segmentos), clasificándolos por su número de lados, la cual puede ser de distinto tamaño según el usuario, como se muestra en la *Figura 14****.***

**Datos de entrada:** posición inicial y final

**Datos de salida:** figura plana realizada (polígono)

**Pre-condiciones:** Posición de partida.



Figura 14. Módulo 1 - Poligonos de Diferentes Tamaños.

***R7:*** El módulo 1 del sistema permite dibujar un polígono de diferentes numeros de lados (3 a 6 lados) dependiendo de la tag utilizada.

**Objetivo:** El módulo permite dibujar una figura plana compuesta por una secuencia de segmentos rectos.

**Descripción:** El módulo del sistema permite trazar el dibujo de una figura plana compuesta por lados (segmentos), clasificándolos por su número de lados (de 3 a 6), el cual se seleccionará dando el tipo de polígono para hacer la figura deseada, como se muestra en la *Figura 15****.***

**Datos de entrada:** posición inicial y final

**Datos de salida:** figura plana realizada (polígono)

**Pre-condiciones:** Posición de partida.

Figura 15. Módulo 1 - Selección del tipo de Poligono.



***R8:*** El módulo 2 del sistema permite la integración con *Kinect*

**Objetivo:** Poder integrar el *Kinect* con ayuda del *Driver* “*OpenNi*” a la computadora y lo reconozca.

**Descripción:** La computadora reconocerá la identificación del *Kinect* con el uso del *Driver* (*OpenNi*), para capturar, utilizar y procesar la información recibida del sensor. *Figura 16****.***

**Datos de entrada:** Información recibida del sensor

**Datos de salida:** Reconocimiento del *Kinect*.

**Pre-condiciones:** Instalación del *Driver (OpenNi)*

.

Figura 16. Módulo 2 - Integración con Kinect



***R9:*** El módulo 2 del sistema permite la detección del dedo índice

**Objetivo:** Poder detectar la imagen que capturará *Kinect*

**Descripción:** Se detectará la imagen, siendo más especifico en este Módulo será el dedo índice que capturará el *Kinect* para poderlo procesar, como se muestra en la *Figura 17****.***

**Datos de entrada:** Información recibida del sensor de *Kinect*

**Datos de salida:** Reconocimiento del dedo índice.

**Pre-condiciones:** Instalación del *API*, *Kinect* conectado

Figura 17. Módulo 2 - Detección del Dedo



Detección

***R10:*** El módulo 2 del sistema permite reconocer el desplazamiento del dedo índice

**Objetivo:** Poder detectar el desplazamiento con el *Kinect*

**Descripción:** Detectar el desplazamiento que hará el dedo índice mediante el *Kinect* la imagen, siendo más especifico en este Módulo será el dedo índice que capturará el *Kinect* para poder ser procesado por la PC, como se muestra en la *Figura 18****.***

**Datos de entrada:** Posición inicial del dedo

**Datos de salida:** Posición final del dedo.

**Pre-condiciones:** Instalación del API, *Kinect* conectado



Reconocer



Figura 18. Módulo 2 - Reconocer Desplazamiento del dedo índice.

***R11:*** El módulo 2 del sistema permite dibujar a mano alzada con el dedo índice

**Objetivo:** Poder plasmar el trazo realizado con el dedo

**Descripción:** Poder ver reflejado la acción del dedo, con la realización del trazo a mano alzada, como se muestra en la *Figura 19****.***

**Datos de entrada:** posición inicial

**Datos de salida:** Trazo realizado

**Pre-condiciones:** Instalación del API, *Kinect* conectado



Figura 19. Módulo 2 - Dibujar a Mano Alzada con el Dedo Índice

***R12:*** El módulo 2 del sistema permite proyectar en el editor básico de dibujo la acción realizada por el dedo índice

**Objetivo:** Poder plasmar el trazo realizado con el dedo en el editor de dibujo

**Descripción:** Poder ver reflejado la acción del dedo, con la realización del trazo a mano alzada en el editor de dibujo básico con la colaboración del *Kinect,* como se muestra en la *Figura 20****.***

**Datos de entrada:** posición inicial

**Datos de salida:** Trazo realizado

**Pre-condiciones:** Instalación del API, *Kinect* conectado

Figura 20. Módulo 2 - Proyección en el editor básico de dibujo.



***R13:*** El módulo 3 del sistema permite trabajar conjuntamente proyector y *Kinect*

**Objetivo:** Poder trabajar simultáneamente el proyector y el *Kinect*

**Descripción:** Tener trabajando colectivamente los dos dispositivos, como se muestra en la *Figura 21,* evitando interferencias (ruido) entre ambos

**Datos de entrada:** Información recibida del sensor del *Kinect*

**Datos de salida:** Reconocimiento de los dispositivos

**Pre-condiciones:** Instalación del API, dispositivos conectados

Figura 21. Módulo 3 - Trabajo Colectivo de los Dispositivos



***R14:*** El módulo 3 del sistema permite proyectar sobre el área de trabajo

**Objetivo:** Poder plasmar la imagen con el proyector sobre el área de trabajo

**Descripción:** Con el proyector se podrá mostrar la imagen en el área de trabajo que se indicará ajustando el dispositivo en una cierta posición, como se muestra en la *Figura 22****.***

**Datos de entrada:** ninguno

**Datos de salida:** Imagen proyectada

**Pre-condiciones:** Instalación del API, *Kinect* y proyector conectado



Figura 22. Módulo 4 - Proyección sobre el área de trabajo.

***R15:*** El módulo 3 del sistema permite trabajar sin interferencia de la sombra que produzca la mano

**Objetivo:** Poder trabajar con la mano, a pesar de la sobra que genere no afectará el producto deseado

**Descripción:** Trabajar con la mano, a pesar de la sombra que genere, no afectará el producto deseado (figura o acción) proyectado sobre el área de trabajo, como se muestra en la *Figura 23****.***

**Datos de entrada:** Procesamiento de imagen

**Datos de salida:** Imagen proyectada

**Pre-condiciones:** Instalación del API, *Kinect* y proyector conectado



Figura 23. Módulo 3 - Trabajar a pesar de la sombra

***R16:*** Elmódulo 4 del sistema permite reconocer la herramienta (*tag*)

**Objetivo:** Que el *Kinect* reconozca la herramienta (*tag*)

**Descripción:** Poder hacer que se reconozca la herramienta mediante el *Kinect*, dicha herramienta la llamamos “*tag*”, la cual es una imagen binaria, como se muestra en la *Figura 24****.***

**Datos de entrada:** imagen binaria

**Datos de salida:** herramienta reconocida

**Pre-condiciones:** Instalación del API, *Kinect* conectado, imagen binaria



TAG

Figura 24. Módulo 4 - Reconocimiento de la herramienta (Tag)

**Nota: la imagen binaria (tag) utilizada es solo un ejemplo**

***R17:*** El módulo 4 del sistema permite dibujar con la herramienta (*tag*) de la figura a usar

**Objetivo:** Se utiliza la herramienta (*tag*) para la acción que tiene prevista, la cual es dibujar

**Descripción:** Poder dibujar alguna de las figuras ya mencionadas, simplemente con el hecho de poner la *tag*, y que los usuarios (máximo 2 usuarios) desplacen el dedo para crear la figura, reconociendo que debe dibujar la figura seleccionada por la *tag* que le corresponde a la figura, como se muestra en la *Figura 25****.***

**Datos de entrada:** imagen binaria

**Datos de salida:** herramienta reconocida

**Pre-condiciones:** Instalación del API, *Kinect* conectado, imagen binaria.



TAG

Figura

***R18:*** El módulo 4 del sistema permite realizar la selección de escalar una figura

Figura 25. Módulo 4 - Dibujar utilizando la Herramienta (tag)

**Objetivo:** Se utiliza la herramienta (*tag*) para la acción que tiene prevista, la cual es cambiar de tamaño la figura

**Descripción:** Poder realizar cambios a alguna de las figuras ya mencionadas, simplemente con el hecho de poner la *tag* y que reconozca que debe cambiar de tamaño la figura seleccionada, girando la *tag* que le corresponde a la acción de escalar, como se muestra la *Figura 26****.***

**Datos de entrada:** imagen binaria

**Datos de salida:** herramienta reconocida

**Pre-condiciones:** Instalación del API, *Kinect* conectado, imagen binaria

Figura 26. Módulo 4 - Escalar Figura.



Figura

TAG

***R19:*** El módulo 4 del sistema permite realizar la selección de mover la figura

**Objetivo:** Se utiliza la herramienta (*tag*) para la acción que tiene prevista, la cual es mover la figura

**Descripción:** Poder realizar cambios a alguna de las figuras ya mencionadas, simplemente con el hecho de poner la *tag* y que reconozca, que debe mover de posición la figura seleccionada por la *tag* que le corresponde a la acción de mover, como se muestra en la *Figura 27****.***

**Datos de entrada:** imagen binaria

**Datos de salida:** herramienta reconocida

**Pre-condiciones:** Instalación del API, *Kinect*  conectado, imagen binaria

Figura 27. Módulo 4 - Mover Figura



Mover

Figura

TAG

***R20:*** El módulo 4 del sistema permite realizar la selección de cambiar el color de la figura

**Objetivo:** Se utiliza la herramienta (*tag*) para la acción que tiene prevista, la cual es cambiar de color la figura

**Descripción:** Poder realizar cambios a alguna de las figuras ya mencionadas, simplemente con el hecho de poner la *tag* y que reconozca que debe cambiar de color la figura seleccionada, girando la *tag* que le corresponde a la acción de cambiar color *figura 28*.

**Datos de entrada:** imagen binaria

**Datos de salida:** herramienta reconocida

**Pre-condiciones:** Instalación del API, *Kinect* conectado, imagen binaria

Figura 28. Módulo 4 - Cambiar de color



Figura

Figura

TAG



TAG

Cambiar color

Color Rojo

Color Azul

# 6. Desarrollo

En este apartado se da paso a explicar cómo fue que se desarrollo el sistema en cada uno de los módulos. Se explica como se hizo el reconocimiento de las Tags y el seguimiento del dedo (handtracking) para poder realizar trazos. Acerca del editor de dibujos se explicará brevemente ya que es un módulo simple y no conlleva mayor problema.

## 6.1 Editor de dibujo básico.

La realización de este módulo no llevo mayor problema ya que el framework que utilizamos, OpenCV, ofrece algunas funciones tanto para el dibujo de ventanas para la interfaz como para el dibujo de líneas y rectangulos que usamos para que se realizaran los trazos.

Para visualizar los trazos se tomaron dos puntos, un inicial y un final, que por medio de estos; para los trazos de linea recta (función cvLine) y rectangulo (función cvRectangle) se pasan como parámetros a las funciones respectivas, mientras que para las demás figuras se toman los mismos dos puntos de inicio-fin y se procede a hacer los calculos descritos en el apartado de análisis y descripción de procesos del módulo uno. De esta manera se determinan puntos que se unen con líneas con las funciones que provee OpenCV: cvPoint y cvLine.

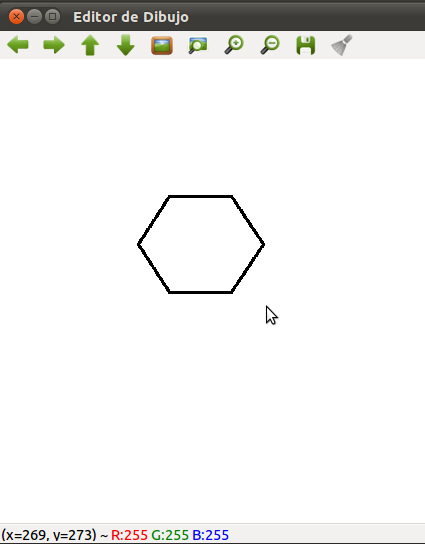
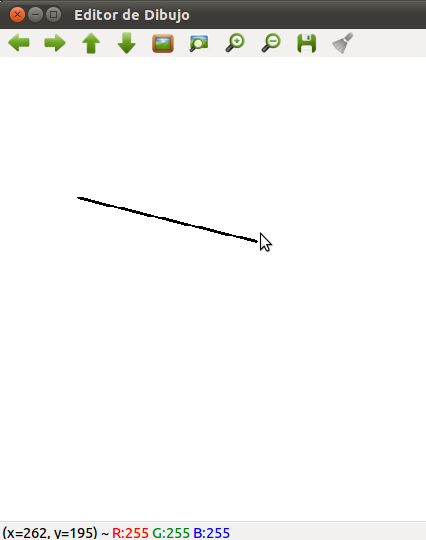


Figura 29. Editor de dibujo básico.

## 6.2 Reconocimiento de trazos a mano alzada.

Para realizar el reconocimiento de los trazos con el kinect se calculan las características de la mano del usuario con los métodos cvMoments y cvHuMoments.

Los momentos son propiedades numéricas que se pueden obtener de una determinada imagen. Tienen en cuenta todos los pixeles de la imagen, no solo los bordes[23][24].

El momento que se utilizó fue el momento central que hace referencia al área (m00).

Una vez calculados los momentos, calculamos los momentos invariantes de Hu, un conjunto de siete momentos invariantes. Estos momentos se mantienen invariantes ante rotaciones, traslaciones y cambios de escalas de objetos. Se define mediante las siguientes ecuaciones[24]. Del cual solo usamos el momento invariante uno correspondiente a la rotación.

Con el momento de área y el primer invariante de Hu (m00 y hu1) comparamos estos datos para poder identificar el dedo índice y conforme a esto se realiza los trazos. Lo que se le llama un reconocimiento de patrones supervisado.

## 6.3 Implementación de herramientas físicas

Este apartado se enfoca en explicar cual fue la tag elegida de las analizadas y cómo se hace el reconocimiento de la misma para la selección de alguna opción del editor de dibujo.

### 6.3.1 Elección de tag

La tag elegida es una combinación del marcador ARToolKit y los reacTIVision fiductials. Esto se eligio así debido a que por una parte los marcadores de ARToolKit son faciles de reconocer porque constan de un marco negro y dentro de este una figura representativa de la acción que se desea realizar evitando la confusión de no saber que herramienta se esta eligiendo como sucedería con el código QR o los reacTIVision fiductials. Por otra parte los segundos marcadores para su identificación utilizan un árbol binario para saber qué acción debe tomar, se mezclaron ya que en las primeras pruebas de ARToolKit solo y con la técnica que utilizamos para el reconocimiento no logramos diferenciar cada tag de otra. Aplicando árboles nos dio otro punto de referencia para comparar y así saber qué tag se colocó.

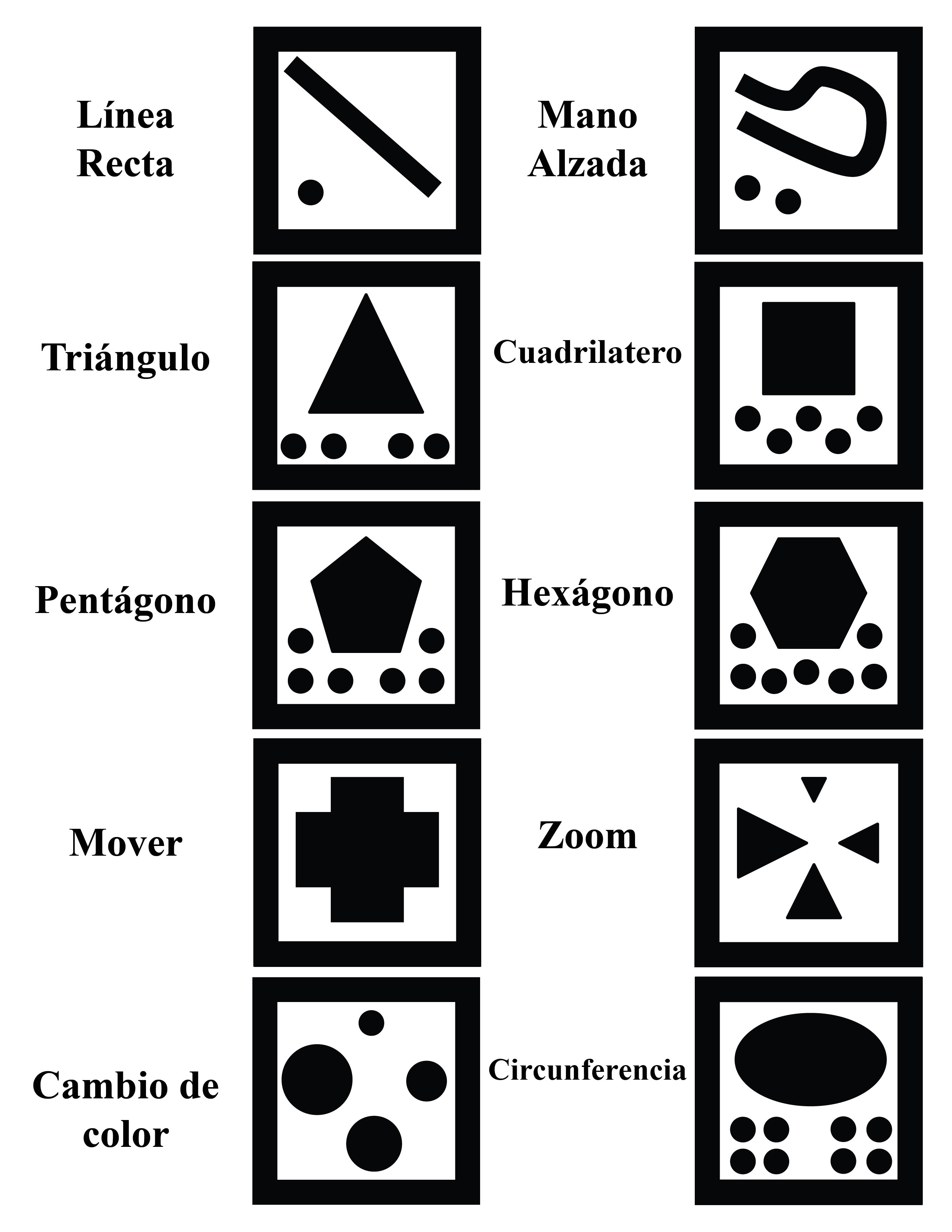


Figura 30 Tag's de herramientas

### 6.3.2 Reconocimiento de Tag’s

Para hacer el reconocimiento de la imagen se realiza lo siguientes procesos.

1. La captura de la imagen es con ayuda del Kinect™, al cual se le indica que la imagen a capturar será del tipo RGB.

2. La imagen capturada la cambiamos a grises con ayuda del método cvCvtColor, el cual multiplica la tonalidad del pixel con los siguientes valores R\*0.299, G\*0.587 y B\*0.114 los cuales se suman y se asignan al mismo pixel [18].

3. La imagen ya en escalas grises se le aplica un suavizado Gaussiano, para realizar este proceso se utilizó el método cvSmooth al cual se le indica que sea de tipo gaussiano con CV\_GAUSSIAN.

El operador de suavizado Gaussiano es un operador de convolución bidimensional que es usado para difuminar imágenes, eliminar detalles y remover el ruido en la imagen. La convolución es realizada por una máscara que representa la función de distribución Gaussiana. La función de distribución Gaussiana unidimensional tiene la forma[19] (1):

(1)

Donde σ es la desviación estándar de la distribución.

En dos dimensiones tenemos la forma (2):

(2)

La idea del suavizado Gaussiano es usar esta distribución bidimensional como una función de “punto de propagación” y esta es llevada a cabo mediante una operación de convolución. Ya que una imagen es almacenada como una colección discreta de pixeles necesitamos realizar una aproximación discreta de la función antes de poder ejecutar la convolución.[19]

El grado de suavizado esta determinado por la desviación estándar σ.

4. Después de suavizar la imagen la Binarizamos, es decir que únicamente tendremos dos colores de la imagen, a Blanco y Negro. Para poder realizar este proceso debemos de calcular un umbral “T” el cual nos permitirá identificar si un pixel cambia a Blanco o Negro(3).[20]

(3)

Para realizar este proceso se utilizó el método cvThreshold del tipo OTSU.

La umbralización es una técnica de segmentación ampliamente utilizada en las aplicaciones

industriales. Se emplea cuando hay una clara diferencia entre los objetos a extraer

respecto del fondo de la escena.[21]

Al aplicar un umbral, ***T***, la imagen en escala de grises, ***f(x,y)***, quedará binarizada; etiquetando con ‘1’ los píxeles correspondientes al objeto y con ‘0’ aquellos que son del fondo.[21]

Una imagen es una función bidimensional de la intensidad del nivel de gris, y contiene

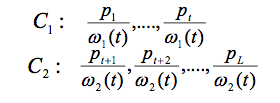
N píxeles cuyos niveles de gris se encuentran entre 1 y L. El número de píxeles con nivel

de gris i se denota como fi, y la probabilidad de ocurrencia del nivel de gris i en la imagen

está dada por (4). [21]

(4)

En el caso de la umbralización en dos niveles de una imagen (a veces llamada binarización), los píxeles son divididos en dos clases: ***C1***, con niveles de gris ***[1, ...., t]***; y ***C2***, con niveles de gris ***[t+1, ...., L]***. Entonces, la distribución de probabilidad de los niveles de gris para las dos clases son[21]:



Donde



La media para la clase C1 y la clase C2



Sea μT la intensidad media de toda la imagen se demuestra que



Otsu definió la varianza entre clases de una imagen umbralizada como



Para una umbralización de dos niveles, Otsu verificó que el umbral óptimo ***t\**** se elige

de manera que σ***B2*** sea máxima; esto es



5.- Después de tener la imagen binarizada, se obtienen todos sus contornos es decir todos

aquellos elementos que son de color blanco dentro de imagen, para obtener todos los contornos y sus elementos dentro del se utiliza el método cvFindContours, este método utiliza el método de contornos de Suzuki[22].

6.- Se calculan las características de cada contorno con los métodos cvMoments y cvHuMoments.

Los momentos son propiedades numéricas que se pueden obtener de una determinada imagen. Tienen en cuenta todos los pixeles de la imagen, no solo los bordes[23][24].

El momento que se utilizó fue el momento central que hace referencia al área (m00).

Una vez calculados los momentos, calculamos los momentos invariantes de Hu, un conjunto de siete momentos invariantes. Estos momentos se mantienen invariantes ante rotaciones, traslaciones y cambios de escalas de objetos. Se define mediante las siguientes ecuaciones[24]. Del cual solo usamos el momento invariante uno.

7.- Con el momento de área y el primer invariante de Hu (m00 y hu1) comparamos estos datos para poder identificar el tipo de Tag. Lo que se le llama un reconocimiento de patrones supervisado.

8.- Mostramos la selección que se realizó de la herramienta.

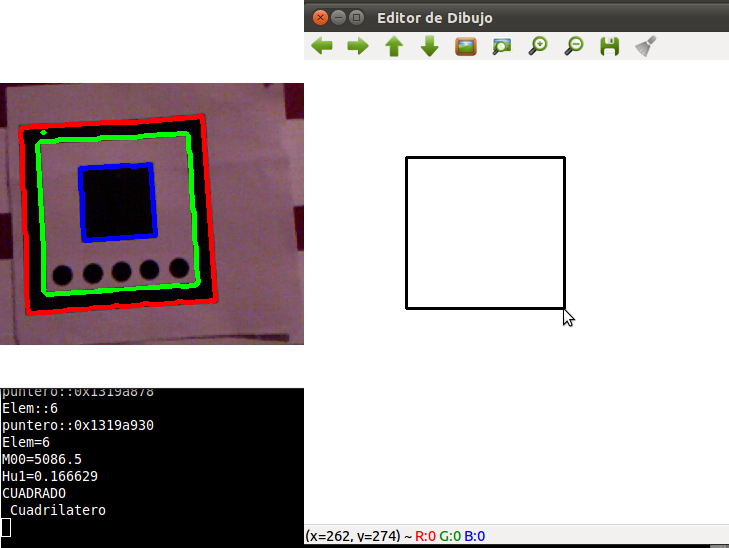


Figura 31 Tag con editor de dibujo.

# 7 Anexo

### 7.1 Diagrama General de Casos de Uso.

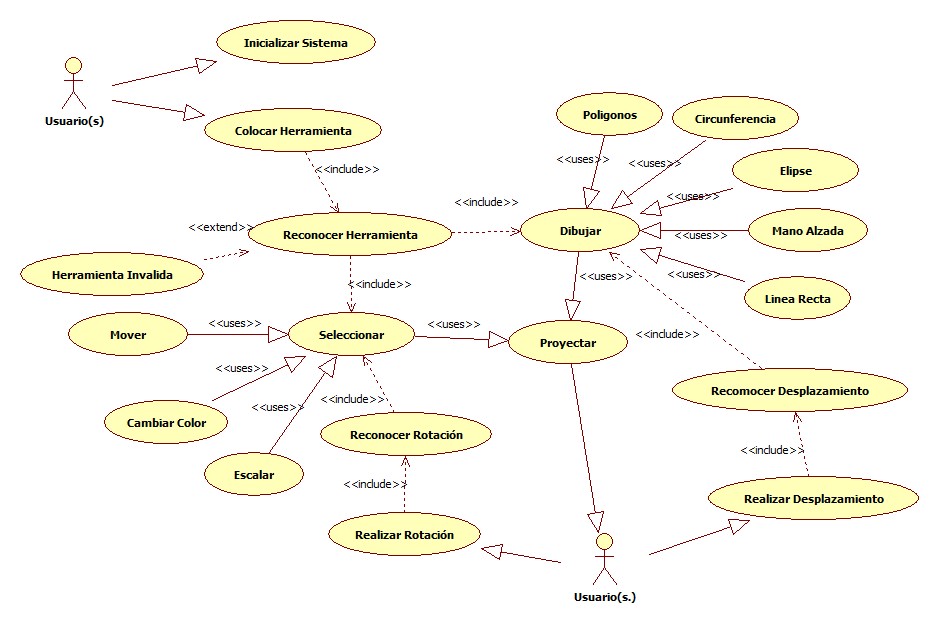


Figura 32. Diagrama General de Casos de Uso

#### 7.1.1 Especificación de casos de uso

Abreviaturas:

E.C.U.: Especificación de Caso de Uso

TT-2011-B007: Trabajo Terminal con No. de Registro 2011-B007

|  |  |
| --- | --- |
| **Especificación del Caso de Uso: Inicializar Sistema** | |
| **ID** | E.C.U-1 |
| **Nombre** | Inicializar el sistema |
| **Descripción** | El usuario inicializara la aplicación del sistema |
| **Autor** | TT-2011-B007 |
| **Actores** | Usuarios (Máximo dos usuarios) |
| **Precondición** | Dispositivos conectados (kinect, proyector),Driver y API instalados |
| **Pos Condición** | Sistema inicializado con éxito |
| **Flujo Normal de eventos** | |
| **Iniciar:**   1. El actor iniciara el sistema de dibujo en 2D multi-touch con reconocimiento de patrones 2. Una vez iniciado, el usuario podrá interactuar con el sistema | |
| **Flujos Alternos** | |
| **Sistema ya iniciado:**   1. Si ya se inicio, se procederá con el paso 2 | |
| **Excepciones** | |
| Ninguna | |
| **Anotaciones** | Ninguna |

|  |  |
| --- | --- |
| **Especificación del Caso de Uso: Colocar Herramienta** | |
| **ID** | E.C.U-2 |
| **Nombre** | Colocar Herramienta |
| **Descripción** | El usuario colocará la herramienta llamada TAG, en el área de trabajo |
| **Autor** | TT-2011-B007 |
| **Actores** | Usuarios (Máximo dos usuarios) |
| **Precondición** | Dispositivo kinect conectado |
| **Pos Condición** | La herramienta estará situada en el área de trabajo |
| **Flujo Normal de eventos** | |
| **Colocación de la Herramienta**   1. El actor seleccionara la herramienta que desea utilizar 2. Se colocara la herramienta en el área de trabajo | |
| **Flujos Alternos** | |
| **Posición de Herramienta:**   1. Después de colocar la herramienta se acomodara en una posición, si se trata de una herramienta para hacer cambios | |
| **Excepciones** | |
| **Herramienta mal posicionada:**   1. Si la herramienta no está del lado indicado ó de la posición correcta, no se podrá trabajar con ella | |
| **Anotaciones** | Solo se puede utilizar una herramienta en el área a la vez |

|  |  |
| --- | --- |
| **Especificación del Caso de Uso: Reconocer Herramienta** | |
| **ID** | E.C.U-3 |
| **Nombre** | Reconocer Herramienta |
| **Descripción** | Cuando ya se encuentre colocada la tag , kinect tomará video del escenario el cual será procesado por la PC y se detecta la tag que este ubicada en el área de trabajo. |
| **Autor** | TT-2011-B007 |
| **Actores** | Usuarios (Máximo dos usuarios) |
| **Precondición** | Dispositivo kinect conectado |
| **Pos Condición** | Quedará habilitada la acción correspondiente a la tag. |
| **Flujo Normal de eventos** | |
| **Detección de imagen**   1. Una vez colocada la herramienta en el área, el kinect captura video del escenario para procesarla 2. Una vez que se procese, se podrá hacer la acción que tiene prevista la herramienta 3. El actor podrá seleccionar ó dibujar en el área de trabajo | |
| **Flujos Alternos** | |
| **Posición de Herramienta:**   1. Si se trata de una herramienta para hacerle cambios a lo dibujado, entonces se moverá de posición la herramienta(giro), para tener dichos cambios, esto solo aplica en escalar y color | |
| **Excepciones** | |
| **Imagen incorrecta:**   1. Si la tag no corresponde con las indicadas en el sistema ,no se podrá reconocer | |
| **Anotaciones** | Solo se puede utilizar una herramienta en el área a la vez |

|  |  |
| --- | --- |
| **Especificación del Caso de Uso: Seleccionar** | |
| **ID** | E.C.U-4 |
| **Nombre** | Seleccionar |
| **Descripción** | Selección del dibujo para poder realizar algún cambio sobre éste. |
| **Autor** | TT-2011-B007 |
| **Actores** | Usuarios (Máximo dos usuarios) |
| **Precondición** | Dispositivo kinect conectado, dibujo previo |
| **Pos Condición** | Se selecciono con éxito el dibujo. |
| **Flujo Normal de eventos** | |
| **Selección:**   1. Ya que se tenga el dibujo, el actor, seleccionará la imagen mediante el uso de su dedo 2. Para por ultimo poder hacer algún cambio en la imagen | |
| **Flujos Alternos** | |
| **Dibujo seleccionado:**   1. Una vez que se está en el paso 1 del flujo normal se puede cambiar el tipo de acción para modificar el dibujo | |
| **Excepciones** | |
| **Selección de dibujo:**   1. Trazos a mano alzada no podrán ser seleccionados. | |
| **Anotaciones** | Solo se puede utilizar una herramienta en el área a la vez , el usuario solo puede utilizar un dedo de la mano. |

|  |  |
| --- | --- |
| **Especificación del Caso de Uso: Herramienta Invalida** | |
| **ID** | E.C.U-5 |
| **Nombre** | Herramienta Invalida |
| **Descripción** | Si se llega a poner otro objeto que no se tenga considerado como herramienta del sistema, entonces el sistema no reconocerá dicha herramienta |
| **Autor** | TT-2011-B007 |
| **Actores** | Usuarios (Máximo dos usuarios) |
| **Precondición** | Dispositivo kinect conectado |
| **Pos Condición** | El objeto es una herramienta invalida para el sistema |
| **Flujo Normal de eventos** | |
| **Herramienta Errónea:**   1. Si el actor llegará a poner otro objeto que no se considero en el sistema, entonces no se podrá reconocer el objeto | |
| **Flujos Alternos** | |
| **Verificar Herramienta:**   1. Si pasara el paso 1 del evento, entonces se verificaría si la herramienta es la correcta o está colocada correctamente. | |
| **Excepciones** | |
| Ninguna | |
| **Anotaciones** | Solo se puede utilizar una herramienta en el área a la vez |

|  |  |
| --- | --- |
| **Especificación del Caso de Uso: Mover** | |
| **ID** | E.C.U-6 |
| **Nombre** | Mover |
| **Descripción** | Esta herramienta nos permite mover de posición alguna figura, para ello debe estar seleccionada |
| **Autor** | TT-2011-B007 |
| **Actores** | Usuarios (Máximo dos usuarios) |
| **Precondición** | Dispositivo kinect conectado. |
| **Pos Condición** | Reconocimiento de la herramienta, poder mover con éxito el dibujo |
| **Flujo Normal de eventos** | |
| **Mover:**   1. Colocara la herramienta de Mover en el área de trabajo 2. El actor que seleccionar el dibujo que desea mover 3. El actor procederá a mover el dibujo seleccionado sobre el área. | |
| **Flujos Alternos** | |
| **ninguno** | |
| **Excepciones** | |
| **Selección de dibujo:**   1. Trazos a mano alzada no podrán ser seleccionados. | |
| **Anotaciones** | Solo se puede utilizar una herramienta en el área a la vez , el usuario solo puede utilizar un dedo de la mano |

|  |  |
| --- | --- |
| **Especificación del Caso de Uso: Cambiar Color** | |
| **ID** | E.C.U-7 |
| **Nombre** | Cambiar Color |
| **Descripción** | Esta herramienta nos permite cambiar de color alguna figura seleccionada |
| **Autor** | TT-2011-B007 |
| **Actores** | Usuarios (Máximo un usuario) |
| **Precondición** | Dispositivo kinect conectado |
| **Pos Condición** | Reconocimiento de la herramienta, poder cambiar de color con éxito el dibujo |
| **Flujo Normal de eventos** | |
| **Color:**   1. Se colocará la herramienta de Cambiar Color en el área de trabajo 2. El actor seleccionará el dibujo que desea cambiar de color 3. Se rotará la herramienta hasta conseguir el color deseado en el dibujo | |
| **Flujos Alternos** | |
| **ninguno** | |
| **Excepciones** | |
| **Selección de dibujo:**   1. Trazos a mano alzada no podrán ser seleccionados. | |
| **Anotaciones** | Solo se puede utilizar una herramienta en el área a la vez , el usuario solo puede utilizar un dedo de la mano |

|  |  |
| --- | --- |
| **Especificación del Caso de Uso: Escalar** | |
| **ID** | E.C.U-8 |
| **Nombre** | Escalar |
| **Descripción** | Esta herramienta nos permite cambiar de tamaño alguna figura seleccionada |
| **Autor** | TT-2011-B007 |
| **Actores** | Usuarios (Máximo un usuario) |
| **Precondición** | Dispositivo kinect conectado |
| **Pos Condición** | Reconocimiento de la herramienta, poder cambiar de tamaño con éxito el dibujo |
| **Flujo Normal de eventos** | |
| **Mover:**   1. Se colocará la herramienta de Escalar en el área de trabajo 2. El actor seleccionará el dibujo que desea cambiar de tamaño 3. Se rotara la herramienta hasta conseguir el tamaño deseado en el dibujo | |
| **Flujos Alternos** | |
| **ninguna** | |
| **Excepciones** | |
| **Selección de dibujo:**   1. Trazos a mano alzada no podrán ser seleccionados. | |
| **Anotaciones** | Solo se puede utilizar una herramienta en el área a la vez , el usuario solo puede utilizar un dedo de la mano |

|  |  |
| --- | --- |
| **Especificación del Caso de Uso: Reconocer Rotación** | |
| **ID** | E.C.U-9 |
| **Nombre** | Reconocer Rotación |
| **Descripción** | El dispositivo kinect transmitirá la información de la escena de la rotación de la tag a la PC, al cambiar de posición (girar) se procesa para aplicar cierto efecto. |
| **Autor** | TT-2011-B007 |
| **Actores** | Usuarios (Máximo un usuario) |
| **Precondición** | Dispositivo kinect , |
| **Pos Condición** | Reconocimiento de la acción de la herramienta |
| **Flujo Normal de eventos** | |
| **Reconocimiento de rotación:**   1. Una vez colocada la herramienta en el área, el kinect captura el escenario para procesarlo. 2. El actor rotara la imagen para obtener la acción designada | |
| **Flujos Alternos** | |
| **Rotación:**   1. Si ya se tiene el paso 1 del evento, entonces se procede a rotar la herramienta las veces que permite el sistema, para obtener la acción designada | |
| **Excepciones** | |
| **Selección de dibujo:**  Ninguna | |
| **Anotaciones** | Solo se puede utilizar una herramienta en el área a la vez , el usuario solo puede utilizar una mano |

|  |  |
| --- | --- |
| **Especificación del Caso de Uso: Realizar Rotación** | |
| **ID** | E.C.U-10 |
| **Nombre** | Realizar Rotación |
| **Descripción** | El dispositivo kinect capturará el escenario y se identificará la rotación de la tag, la cual al cambiar de posición, podremos visualizar la acción designada. |
| **Autor** | TT-2011-B007 |
| **Actores** | Usuarios (Máximo un usuario) |
| **Precondición** | Dispositivo kinect , |
| **Pos Condición** | Reconocimiento de cambio de la herramienta |
| **Flujo Normal de eventos** | |
| **Reconocimiento de rotación:**   1. Giro de la tag por parte del usuario. 2. Se aplica el cambio al dibujo de acuerdo a la acción designada para la tag. | |
| **Flujos Alternos** | |
| **Rotación:**   1. Giro incompleto de la tag. | |
| **Excepciones** | |
| **Selección de dibujo:**   1. Trazos a mano alzada no podrán ser seleccionados. 2. La herramienta tiene un limite de giros. | |
| **Anotaciones** | Solo se puede utilizar una herramienta en el área a la vez , el usuario solo puede utilizar un dedo de la mano. |

|  |  |
| --- | --- |
| **Especificación del Caso de Uso: Proyectar** | |
| **ID** | E.C.U-11 |
| **Nombre** | Proyectar |
| **Descripción** | El proyector nos dará la visualización del área de trabajo |
| **Autor** | TT-2011-B007 |
| **Actores** | Usuarios (Máximo dos usuarios) |
| **Precondición** | Dispositivos: kinect y proyector conectados |
| **Pos Condición** | Proyección del área de trabajo |
| **Flujo Normal de eventos** | |
| **Proyección:**   1. Una vez iniciado el sistema se proyectará el área de trabajo 2. El usuario puede empezar a dibujar o colocar herramientas para dibujar. | |
| **Flujos Alternos** | |
| Ninguno | |
| **Excepciones** | |
| Ninguno | |
| **Anotaciones** | Al inicio del sistema solo se visualizara el área de trabajo. |

|  |  |
| --- | --- |
| **Especificación del Caso de Uso: Reconocer Desplazamiento** | |
| **ID** | E.C.U-12 |
| **Nombre** | Reconocer Desplazamiento |
| **Descripción** | El dispositivo kinect capturará el desplazamiento del dedo para que lo procese la PC. |
| **Autor** | TT-2011-B007 |
| **Actores** | Usuarios (Máximo dos usuarios) |
| **Precondición** | Dispositivos: kinect conectado |
| **Pos Condición** | Reconocimiento del dedo con éxito |
| **Flujo Normal de eventos** | |
| **Proyección:**   1. El actor realizará el desplazamiento de su dedo. 2. El kinect capturará la escena y la envía a la PC para procesar el movimiento que se realice. | |
| **Flujos Alternos** | |
| Ninguno | |
| **Excepciones** | |
| Solo se usara el dedo índice | |
| **Anotaciones** | Los usuarios solo pueden usar un solo dedo índice por usuario |

|  |  |
| --- | --- |
| **Especificación del Caso de Uso: Realizar Desplazamiento** | |
| **ID** | E.C.U-13 |
| **Nombre** | Realizar Desplazamiento |
| **Descripción** | El actor hará el desplazamiento del dedo hasta la posición que se desea |
| **Autor** | TT-2011-B007 |
| **Actores** | Usuarios (Máximo dos usuarios) |
| **Precondición** | ninguna |
| **Pos Condición** | Reconocimiento del dedo con éxito |
| **Flujo Normal de eventos** | |
| **Proyección:**   1. El actor dezplazará su dedo por el área de trabajo de un punto a otro. 2. El kinect captura y la PC procesara el movimiento que realice el dedo | |
| **Flujos Alternos** | |
| Ninguno | |
| **Excepciones** | |
| Solo se usara el dedo índice | |
| **Anotaciones** | Los usuarios solo pueden usar un dedo por usuario |

|  |  |
| --- | --- |
| **Especificación del Caso de Uso: Dibujar** | |
| **ID** | E.C.U-14 |
| **Nombre** | Dibujar |
| **Descripción** | El actor podrá realizar trazos dentro del área de trabajo. |
| **Autor** | TT-2011-B007 |
| **Actores** | Usuarios (Máximo dos usuarios) |
| **Precondición** | Dispositivos: kinect y proyector, tags en caso de requerirlo. |
| **Pos Condición** | Visualización de trazo realizado |
| **Flujo Normal de eventos** | |
| **Dibujar con herramienta:**   1. El actor pondrá la herramienta de la figura deseada 2. Desplazara su dedo índice para formar la figura | |
| **Flujos Alternos** | |
| **Dibujar sin herramienta:**   1. Si el actor quiere dibujar un trazo a mano alzada simplemente realizara el trazo sin tag en el área de trabajo. | |
| **Excepciones** | |
| Solo se usará el dedo índice para realizar el trazo | |
| **Anotaciones** | Los usuarios solo pueden usar un dedo por usuario |

|  |  |
| --- | --- |
| **Especificación del Caso de Uso: Mano Alzada** | |
| **ID** | E.C.U-15 |
| **Nombre** | Mano alzada |
| **Descripción** | El actor podrá realizar ciertos trazos a mano alzada dentro del área de trabajo |
| **Autor** | TT-2011-B007 |
| **Actores** | Usuarios (Máximo dos usuarios) |
| **Precondición** | Dispositivos: kinect y proyector , Posición inicial |
| **Pos Condición** | Trazo realizado |
| **Flujo Normal de eventos** | |
| **Dibujar a mano alzada:**   1. El actor usara su dedo con un movimiento libre de un punto inicial a un punto final, dentro del área de trabajo 2. El kinect captura la escena y se reflejará el trazo realizado | |
| **Flujos Alternos** | |
| Ninguno | |
| **Excepciones** | |
| **Velocidad de desplazamiento:**   1. Si la velocidad de desplazamiento varía de la velocidad de procesamiento, el trazo realizado podría no ser igual al movimiento del dedo. | |
| **Anotaciones** | Los usuarios solo pueden usar un solo dedo índice por usuario |

|  |  |
| --- | --- |
| **Especificación del Caso de Uso: Línea Recta** | |
| **ID** | E.C.U-16 |
| **Nombre** | Línea Recta |
| **Descripción** | El actor podrá realizar un trazo recto a partir de dos puntos |
| **Autor** | TT-2011-B007 |
| **Actores** | Usuarios (Máximo dos usuarios) |
| **Precondición** | Dispositivos: kinect y proyector , Posición inicial |
| **Pos Condición** | Trazo realizado |
| **Flujo Normal de eventos** | |
| **Dibujar a mano alzada:**   1. El actor pondrá la herramienta para lineas 2. El actor usara su dedo con un movimiento libre dentro del área de trabajo 3. El kinect capturará la escena y se reflejará el trazo realizado en el área | |
| **Flujos Alternos** | |
| **Interacción de los actores:**   1. El desplazamiento que realiza el actor ,puede variar con lo que se dibuja | |
| **Excepciones** | |
| **Segmentación de Pixel:**  Se puede apreciar una pequeña deformación en la línea, al no poder segmentar un pixel. | |
| **Anotaciones** | Los usuarios solo pueden usar un dedo por usuario |

|  |  |
| --- | --- |
| **Especificación del Caso de Uso: Elipse** | |
| **ID** | E.C.U-17 |
| **Nombre** | Elipse |
| **Descripción** | El actor podrá dibujar una curva plana y cerrada, simétrica respecto a dos ejes perpendiculares entre sí. |
| **Autor** | TT-2011-B007 |
| **Actores** | Usuarios (Máximo dos usuarios) |
| **Precondición** | Dispositivos: kinect y proyector , Posición inicial |
| **Pos Condición** | Elipse realizada |
| **Flujo Normal de eventos** | |
| **Dibujar de elipse:**   1. El actor pondrá la herramienta para el elipse 2. El actor usara su dedo con un movimiento libre dentro del área de trabajo 3. El kinect lo capturará y se reflejara el trazo realizado en el área | |
| **Flujos Alternos** | |
| **Interacción de los actores:**   1. Cuando se encuentre el paso 1, los actores podrán realizar su trazo respectivamente dentro del área de trabajo | |
| **Excepciones** | |
| **Distancia de puntos de la figura:**   1. Se puede apreciar una deformación mayor cuando la distancia entre los puntos sea menor, al ser pixeles de forma cuadrada. | |
| **Anotaciones** | Los usuarios solo pueden usar un dedo por usuario |

|  |  |
| --- | --- |
| **Especificación del Caso de Uso: Circunferencia** | |
| **ID** | E.C.U-18 |
| **Nombre** | Circunferencia |
| **Descripción** | El actor podrá dibujar una superficie plana limitada por una circunferencia |
| **Autor** | TT-2011-B007 |
| **Actores** | Usuarios (Máximo dos usuarios) |
| **Precondición** | Dispositivos: kinect y proyector , Posición inicial |
| **Pos Condición** | Circunferencia realizada |
| **Flujo Normal de eventos** | |
| **Dibujar de elipse:**   1. El actor pondrá la herramienta para la circunferencia 2. El actor usara su dedo con un movimiento libre dentro del área de trabajo 3. El kinect lo capturará y se reflejara el trazo realizado en el área | |
| **Flujos Alternos** | |
| **Interacción de los actores:**   1. Cuando se encuentre el paso 1, los actores podrán realizar su trazo respectivamente dentro del área de trabajo | |
| **Excepciones** | |
| **Distancia de puntos de la figura:**   1. Se puede apreciar una deformación mayor cuando la distancia entre los puntos sea menor, al ser pixeles de forma cuadrada. | |
| **Anotaciones** | Los usuarios solo pueden usar un dedo por usuario |

|  |  |
| --- | --- |
| **Especificación del Caso de Uso: Polígono** | |
| **ID** | E.C.U-19 |
| **Nombre** | Polígono |
| **Descripción** | El actor podrá dibujar una figura plana compuesta por una secuencia de segmentos rectos , que se puede elegir entre 3 a 6 lados |
| **Autor** | TT-2011-B007 |
| **Actores** | Usuarios (Máximo dos usuarios) |
| **Precondición** | Dispositivos: kinect y proyector , Posición inicial |
| **Pos Condición** | Polígono realizado |
| **Flujo Normal de eventos** | |
| **Dibujar de elipse:**   1. El actor pondrá la herramienta del polígono deseado. 2. El actor usara su dedo con un movimiento libre dentro del área de trabajo 3. El kinect lo capturará y se reflejara el trazo de la circunferencia | |
| **Flujos Alternos** | |
| **Interacción de los actores:**   1. Cuando se encuentre el paso 1, los actores podrán realizar su trazo respectivamente dentro del área de trabajo | |
| **Excepciones** | |
| **Dimensión de la figura:**   1. Se puede apreciar una deformación, cuando se le da una dimensión, al no poder segmentar un pixel. 2. Solo hay tags para poligonos de 3 a 6 lados. | |
| **Anotaciones** | Los usuarios solo pueden usar un dedo por usuario |

### 7.2 Diagrama General de Clases

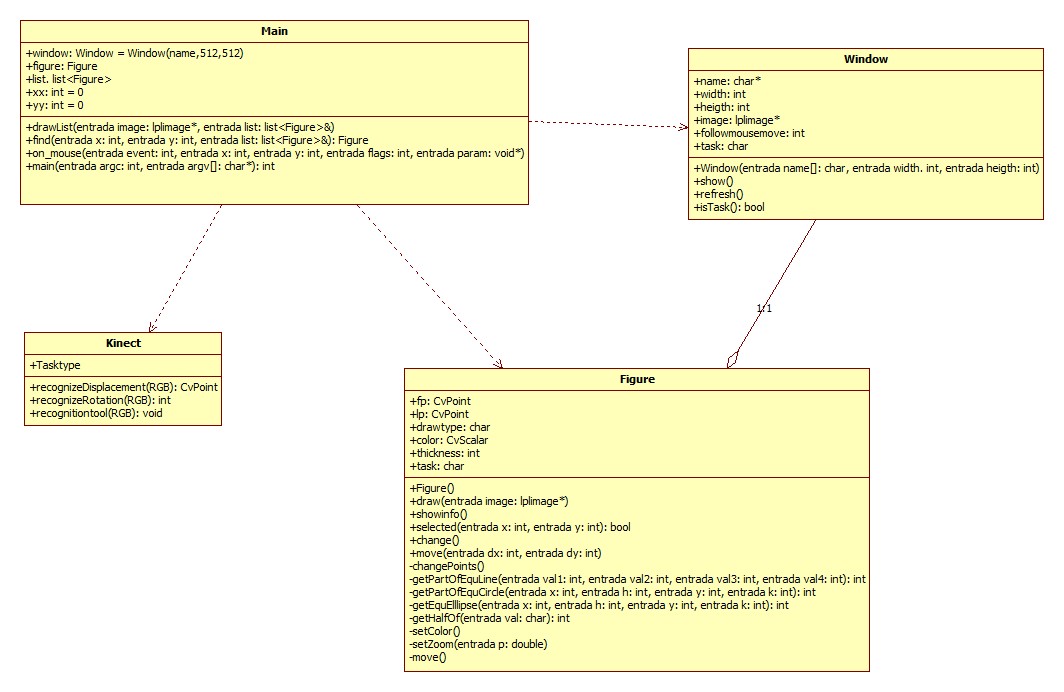


Figura 33. Diagrama General de Clases

### 7.3 Diagrama General de Secuencia

Figura 34. Diagrama General de Secuencia.

### 7.4 Diagrama General de Estados

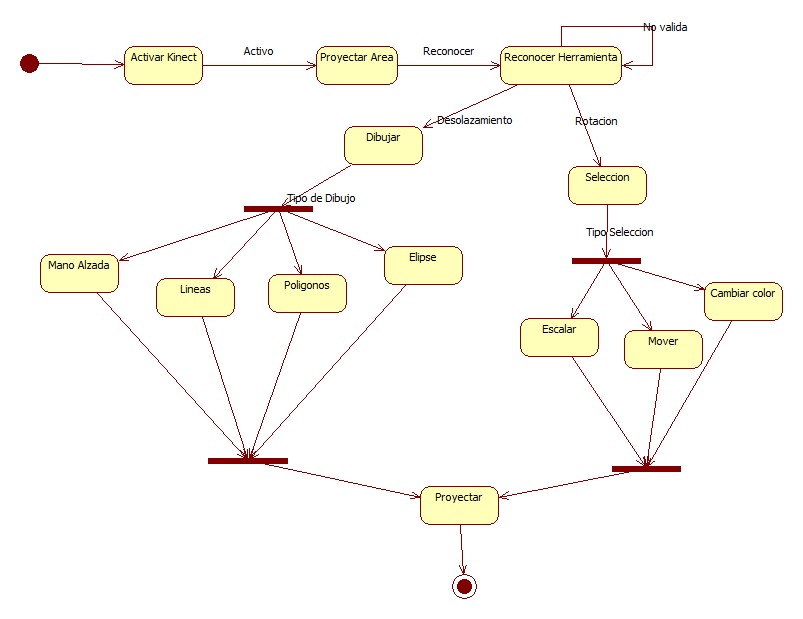


Figura 35. Diagrama General de Estados

### 7.5 Diagrama de Casos de Uso Módulo 1

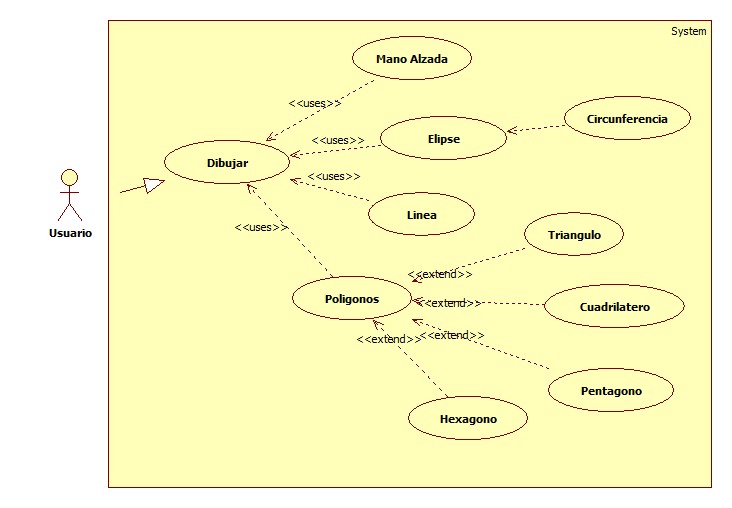


Figura 36. Diagrama de Casos de Uso - Módulo 1.

### 7.6 Diagrama de clases Módulo 1

### 

Figura 37 Diagrama de Clases - Módulo 1.

### 7.7 Diagrama de Secuencia Módulo 1

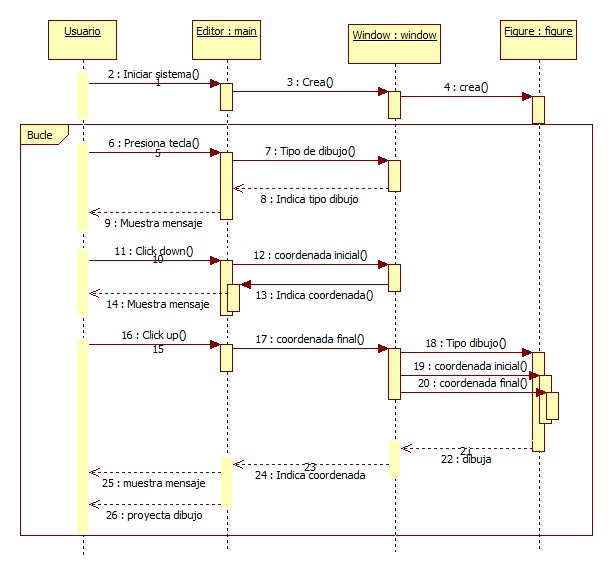
******

Figura 38. Diagrama de Secuencia - Módulo 1.

### 7.8 Diagrama de Estados Módulo 1

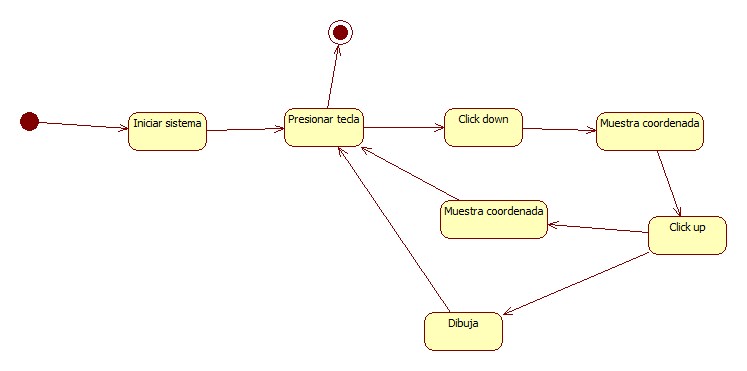


Figura 39. Diagrama de Estados - Módulo 1.

### 7.9 Diagrama de Casos de Uso Módulo 2

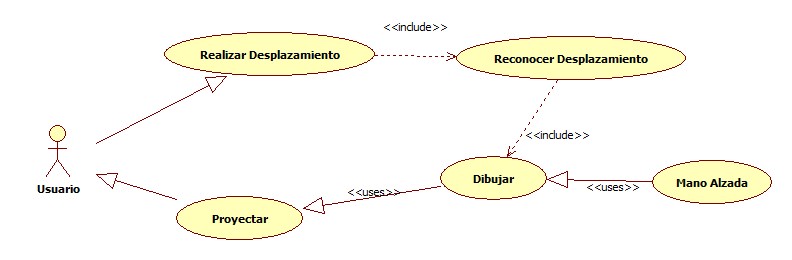


Figura 40. Diagrama de Casos de Uso - Módulo 2.

### 7.10 Diagrama de Clases Módulo 2

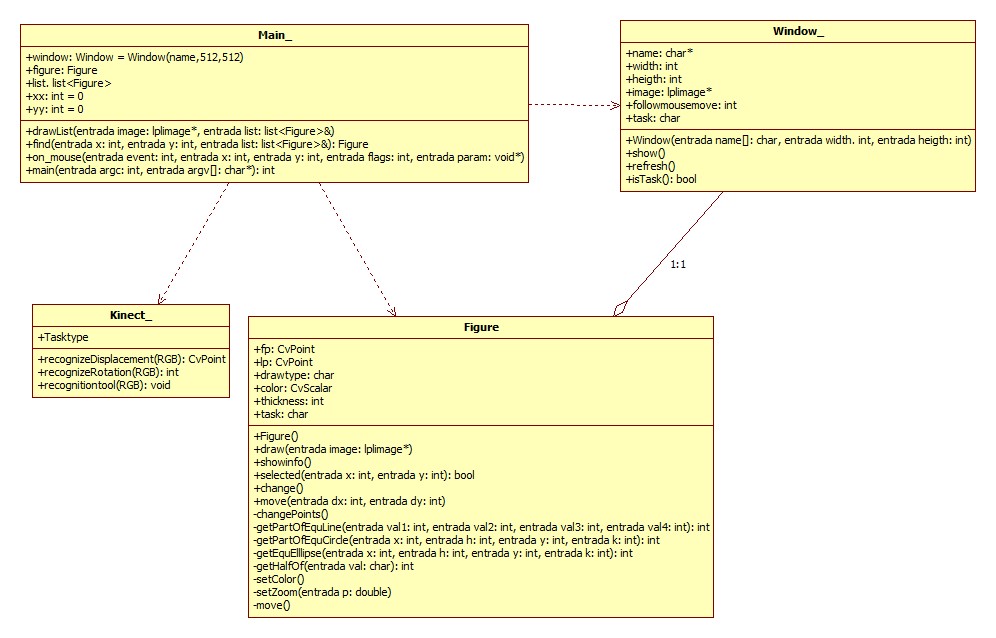


Figura 41. Diagrama de Clases - Módulo 2.

### 7.11 Diagrama de Secuencias Módulo 2

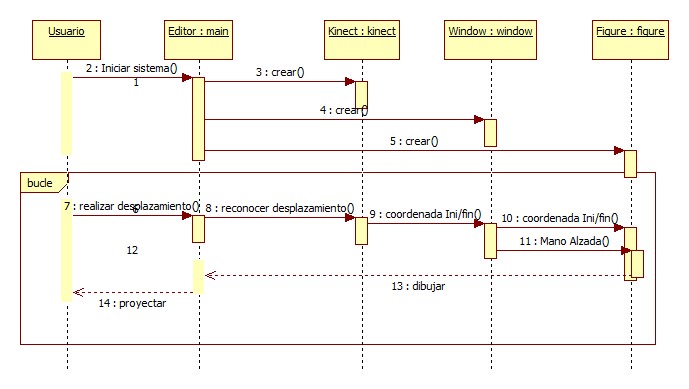


Figura 42. Diagrama de Secuencia - Módulo 2.

### 7.12 Diagrama de Estados Módulo 2



Figura 43. Diagrama de Estados - Módulo 2.

### 7.13 Diagrama de Casos de Uso Módulo 3

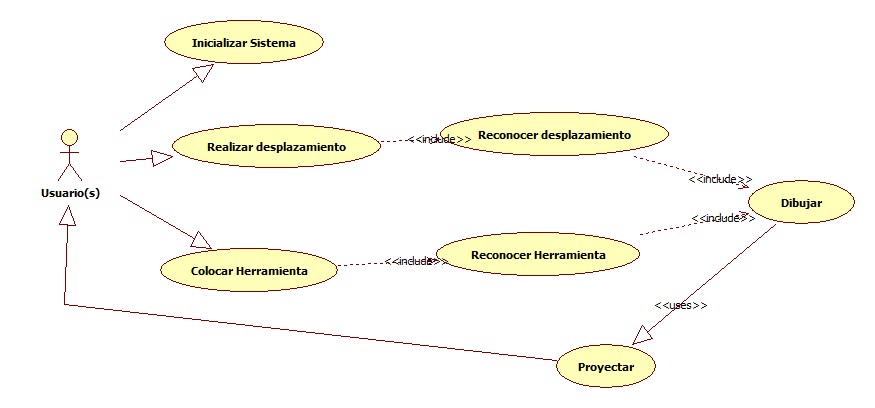


Figura 44. Diagrama de Casos de Uso - Módulo 3.

### 7.14 Diagrama de Secuencia Módulo 3.

### 

Figura 45. Diagrama de Secuencia - Módulo 3.

### 7.15 Diagrama de Estados Módulo 3

### 

Figura 46. Diagrama de Estados - Módulo 3.

### 7.16 Diagrama de Casos de Uso Módulo 4.

### 

Figura 47. Diagrama de Casos de Uso - Módulo 4.

### 7.17 Diagrama de Clases Módulo 4

### 

Figura 48. Diagrama de Clases - Módulo 4.

### 7.18 Diagrama de Secuencia Módulo 4.

### 

Figura 49. Diagrama de Secuencia - Módulo 4.

### 7.19 Diagrama de Estados Módulo 4.

### 

Figura 50. Diagrama de Estados - Módulo 4.

# 8. Referencias.

[1] Online. Systems, Intelligence, Robotics and Perception, Pontificia Universidad Javeriana Bogotá http://www.gruposirp.org/sirp/wiki/doku.php http://gruposirp.org/sirp/sirp-home.html WebSite 2011.

[2] Presentación Kinect Bruno Capuano Visual Studio ALM

https://mvp.support.microsoft.com/profile=800D1522-8788-4A34-8985-233DBBF2A40A Conference Cidetec IPN 2011 cortesía Ing. Naím Rivera.

[3] Online. OpenNI™. <http://75.98.78.94/About.aspx>

[4] Open Source Computer Vision Library.Reference Manual. Edic. Intel Corporation.

[5] “Pattern Recognition” Sergios Theodoquidis, Konstantinos Kourtroumbus Ed. Acaemic Press 2° Edición 2003.

[6] “Reconocimiento de Patrones: Enfoque Lógico Combinatorio” José Ruiz‐Shulcloper Ed. Instituto Politécnico Nacional.

[7] “The Design and Evolution of Fiducials for the reacTIVision System”,Ross Bencina and Martin Kaltenbrunner, Universitat Pompeu Fabra, Barcelona, España, 2005.

[8] Online. reacTIVison, <http://reactivision.sourceforge.net/>

[9] Online. QR code, versions 1 - 40. <http://www.denso-wave.com/qrcode/vertable1-e.html>

[10] Online. Otsu Thresholding, The Lab bookpages <http://www.labbookpages.co.uk/software/imgProc/otsuThreshold.html>

[11] Online. ZXing web page. <http://code.google.com/p/zxing/>

[12] Online. ARToolkit Documentation. <http://www.hitl.washington.edu/artoolkit/documentation/userarwork.htm>

[13] Roger S. Pressman (adaptado por Darrel Ince), Ingeniería del Software un enfoque práctico, 5ta ed. Ed. McGraw-Hill, pp. 23-24.

[14]Síndrome del túnel carpiano **|** *Causas y factores de riesgo* <http://familydoctor.org/familydoctor/es/diseases-conditions/carpal-tunnel-syndrome/causes-risk-factors.html>

[15] Online. Hand Tracking (Kinect with OpenCV & OpenNI). <http://www.youtube.com/watch?v=qNH_MqqOPX0>

[16] Online. Kinect Active Projection Mapping. <http://www.youtube.com/watch?v=hkHUGxP3ecI>

[17] Online. Technical University Bergakademie Freiberg <http://www.informatik.tu-freiberg.de/>

[18] Online. Manual de referencia OpenCV. <http://www.seas.upenn.edu/~bensapp/opencvdocs/ref/opencvref_cv.htm>

[19] J. R. Parker., Algorithms for Image Processing and Computer Vision, Wiley,1997.

[20] Online. “Técnicas de procesamiento digital de imágenes y reconocimiento de patrones.” <http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/nieto_b_d/capitulo2.pdf>

[21] Online “Segmentación por Umbralización”, Universidad Nacional de Quilmes, 2005

<http://iaci.unq.edu.ar/materias/vision/archivos/apuntes/Segmentaci%C3%B3n%20por%20umbralizaci%C3%B3n%20-%20M%C3%A9todo%20de%20Otsu.pdf>

[22] “Optimizing Connected Component Labeling Algorithms” Kensheng Wu, Ekow Otoo and Arie Shoshani, Lawrence Berkeley National Laboratory, Berkeley, CA, USA.

[23] “Image Description Using Moments”, Dr. S. Belkasim.

[24] “Momentos Wavelet del Tipo B-Spline Cùbicos para Clasificación de Objetos”, Judith Pérez Marcial, Benemérita Universidad Autónoma de Puebla, 2005.