

Universidad Autónoma Metropolitana

Unidad Azcapotzalco

División de Ciencias Básicas e Ingeniería
Ingeniería en Computación

Reporte final de Proyecto Terminal:

**“CLASIFICADOR DE OBJETOS EN BANDA INFINITA POR
MEDIO DE PROCESAMIENTO DIGITAL DE IMÁGENES”**

Alumno: Díaz Cabrera Fausto Mario.

Matrícula: 205202316.

Trimestre: 09-P.

Julio 2009.

Asesor: Dr. Risto Fermín Rangel Kuoppa.

Numero económico: 27499.

1. Introducción.

El proyecto realizado pretende la identificación de un objeto en una banda infinita por medio del procesamiento digital de imágenes, las cuales son captadas por medio de una cámara web. El programa es el encargado de la manipulación digital de diversas imágenes para con ello, lograr el reconocimiento de los objetos.

2. Objetivo General.

Con base en el objetivo general que fue: *“Diseñar e implementar un sistema que clasifique objetos conforme éstos avanzan sobre una banda infinita; la clasificación se hará con la generación de descriptores obtenidos a través del procesamiento digital de imágenes”*.

Se logro que el programa pudiera identificar determinados objetos basándose en su color y forma, usando una cámara web para capturarlos, se utilizaron algoritmos de manipulación de imágenes para lograr obtener un descriptor del objeto y por medio de comparación lograr identificarlo.

Este objetivo se obtuvo gracias a la elaboración de 5 módulos los cuales se encargan de una tarea específica, cada uno de ellos requiere el cumplimiento previo del modulo anterior para su total función.

El modulo número 1, se encarga de obtener un *stream de video*¹ de la cámara web, este recibe todo el contenido que la cámara web puede percibir. En este mismo modulo se puede obtener la imagen que será el fondo para las siguientes comparaciones.

El modulo número 2, tiene como cometido la captura de una imagen precisa que resulta en una imagen la cual es usada por el modulo 3 para realizar su función.

El modulo número 3, es el encargado de realizar la resta de las imágenes, teniendo como entrada la imagen obtenida del modulo 2 y la imagen de fondo obtenida en el modulo 1, su salida es una imagen binaria, que está compuesta por fondo negro, y objeto en blanco.

¹ Tipo de tecnología que permite almacenar en un búfer lo que se ve en la webcam sin necesidad de descargar ni ocupar espacio en disco duro.

El modulo número 4, tiene la función de obtener el contorno del objeto a identificar, su entrada es la imagen binaria del modulo 3 y su salida es una imagen que contiene un fondo en negro y el contorno de la figura en blanco.

El modulo número 5, obtiene un descriptor del objeto, teniendo como base la imagen del modulo 4, recorta esta imagen para obtener otra, compuesta por el objeto y la menor cantidad de fondo, una vez realizado esto, genera un descriptor del objeto, que es comparado en la base de datos, para así por medio de comparación, determinar de qué figura se trata.

Estos módulos fueron sincronizados usando un ciclo *while*² y teniendo variables de retorno booleanas³ en los métodos invocados, esto da como resultado que si el método invocado hace su tarea con éxito, el método siguiente podrá realizar su trabajo y así sucesivamente, en caso contrario el método detendrá el ciclo y este se volverá a realizar hasta que todo cumplan con su objetivo.

Además de estos 5 módulos principales se agrego un modulo extra el cual permite introducir objetos a la base de datos, este sigue el mismo procedimiento que el de identificar un objeto, solo que no realiza comparaciones, simplemente agrega el descriptor del objeto a una tabla hash.

El objetivo general de este proyecto terminal, se cumple ya que es capaz de realizar todos y cada uno de los módulos logrando así la identificación de los objetos deseados.

² El Bucle while o bucle mientras es una estructura de la mayoría de los lenguajes de programación, estructurados cuyo propósito es repetir un bloque de código mientras una condición se mantenga verdadera.

³ El tipo de dato lógico o booleano es en computación aquel que puede representar valores de lógica binaria, esto es, valores que representen falso o verdadero

3. Objetivos Particulares.

Involucrando cada uno de los objetivos particulares, se obtuvo un resultado satisfactorio del proyecto. Se describe cada uno de ellos, haciendo hincapié en las fortalezas y debilidades que se encontraron en la paulatina realización de estos, tomando en cuenta que sin el conjunto y plena función de ellos este proyecto terminal no hubiese cumplido su fin.

1. Diseñar e implementar un módulo que permita la recepción de un stream de video y separar cuadros (imágenes estáticas, instantáneas) de dicho stream.

Para la realización de este modulo se tuvo que dividir en tres partes una que es la encargada de recolectar la información de la cámara web, otra que manipula esa información y obtiene una imagen estática instantánea y finalmente la encargada de guardar una imagen como fondo. Aunque en realidad se trata de dos módulos (Módulo número 1 encargado del *stream de video*. Módulo número 2 que obtiene una imagen estática instantánea), para fines prácticos y de funcionamiento total del proyecto se trataran y explicaran como uno solo.

El diseño de este modulo fue con base en dos *PictureBox*⁴, encargados de la recolección tanto del stream de video como de las imágenes estáticas. Un *ComboBox*⁵, donde se muestra y selecciona los dispositivos capaces de transmitir video a nuestro modulo. Tres *Button*⁶, los cuales realizan la función de comenzar a capturar el video, detener la transmisión de video y guardar la imagen de fondo.

Utilizando el *API*⁷ de *Windows*⁸, fue como se logro capturar el video proveniente de la cámara web, para que esta se mostrara como un *stream de video*. La captura de la instantánea, simplemente se solicita que en un instante dado se obtenga la imagen que en ese momento está siendo transmitida hacia el *PictureBox*, y sea retransmitida hacia el siguiente *PictureBox*. Para lograr guardar una imagen de fondo, ocurre el mismo procedimiento anterior solo que esta vez no es transmitida hacia el *PictureBox*, sino que la imagen es almacenada. Este modulo está compuesto por tres clases:

⁴ Representa un control de cuadro de imagen de Windows para mostrar una imagen.

⁵ El control *ComboBox* de los formularios *Windows Forms* se utiliza para mostrar datos en un cuadro combinado desplegable

⁶ Representa un control de botón de Windows.

⁷ *Application Programming Interface*.- conjunto de procedimientos, métodos, clases o protocolos que un sistema operativo, librería o servicio provee para ser utilizado por otro software como una capa de abstracción.

⁸ Sistema operativo desarrollado por Microsoft ®.

1. La encargada de mostrar el video, tiene como entrada la información procedente de la cámara web, y da como resultado un video continuo y uniforme sobre un *PictureBox*. Además se encarga de listar y seleccionar el dispositivo del cual se desea capturar la información.
2. Su función es capturar una imagen estática del *stream de video*, y enviarlo como la salida de este modulo hacia la entrada del siguiente modulo, para su futuro análisis.
3. Captura la imagen estática del *PictureBox* donde está el video y lo guarda como una imagen, para su futuro análisis.

2. Elegir un algoritmo e implementar el modulo que realice resta de imágenes⁹.

El algoritmo de *resta de imágenes*, se utilizó en el proyecto de la siguiente manera:

1. Se utiliza la imagen guardada como fondo de módulo anterior. Una imagen que funcionara como fondo, no debe de contener ningún objeto que se desee identificar, así como no deberá de modificarse mientras se desee identificar más objetos dentro del mismo marco de referencia.
2. La imagen estática instantánea es la que fue obtenida en el modulo anterior, esta deberá contener el objeto a identificar, existiendo este objeto como única diferencia entre la imagen llamada fondo y esta.
3. Se recorren las dos imágenes (imagen fondo e imagen con objeto) *pixel*¹⁰ a *pixel* obteniendo de cada uno las componentes *RGB*¹¹, si el resultado está dentro del parámetro de tolerancia entonces se trata de un pixel que pertenece al fondo, (a este pixel se le asigna un color negro). Si el resultado es mayor que el nivel de tolerancia, se llega a la conclusión de que los pixeles son parte del objeto, (a este pixel se le asigna un color blanco).

Como resultado se obtiene una imagen binaria del objeto, apareciendo el fondo negro y todo el objeto que se identificara en blanco. Esta imagen será usada por el siguiente modulo para obtener un contorno del objeto.

Para que este modulo pueda realizar una *resta de imágenes* más precisa y pueda ser utilizada efectivamente por el modulo siguiente, fue necesario incluir un

⁹ Procedimiento por el cual se detecta la diferencia entre dos imágenes.

¹⁰ Menor unidad homogénea en color que forma parte de una imagen digital.

¹¹ Color hace referencia a la composición del color en términos de la intensidad de los colores primarios con que se forma: el rojo, el verde y el azul.

*TrackBar*¹², este *TrackBar* informa al módulo que tanto se desea que sea tolerante al realizar la *resta de imágenes*, una tolerancia menor puede dar como resultado que no solo el objeto sea pintado de blanco si no muchos *pixeles* mas, ya que al realizar la resta y haber una pequeña variación de color por muy pequeña que esta sea el programa representara esto como que se trata de una parte que no estaba en el fondo así que le asignara un color blanco, esto nos llevara a obtener un mal contorno de la imagen en el módulo siguiente y por lo tanto una identificación errónea o nula del objeto. Es necesario que lo único que aparezca en blanco sea el objeto, y ningún otro pixel mas, para esto se pueden realizar algunas pruebas antes de comenzar a identificar objetos, ajustando el nivel deseado de tolerancia.

3. Elegir un algoritmo e implementar el módulo para obtener el contorno del objeto que se desea identificar.

Este módulo tiene como entrada la imagen binaria elaborada por el modulo anterior, cabe hacer notar y es muy importante que en esta imagen lo único que se encuentre en blanco sea el objeto a identificar ya que si esto no sucede alterara por completo la identificación del objeto.

El algoritmo que se eligió para obtener el contorno de los objetos a identificar es el siguiente que se describe:

Se comienza por hacer un recorrido *pixel a pixel* de la imagen, cada uno de ellos se tomara como *pixel* base y se obtiene información de sus 4 *pixeles* vecinos (derecha, izquierda, arriba y abajo), tomando en cuenta que es una imagen binaria la cual solo contiene *pixeles* blancos (objeto) y *pixeles* negros (fondo), se tienen dos casos:

1. El *pixel* base y sus 4 *pixeles* vecinos son del mismo color entonces se trata o de un *pixel* que pertenece al fondo o pertenece al interior del objeto, por lo cual se le asigna un color negro.
2. El *pixel* base y alguno de los 4 *pixeles* vecinos son del diferente color entonces se trata de un *pixel* que se encuentra en el contorno del objeto, a este *pixel* se le asigna un color blanco.

Una vez encontrado el contorno del objeto esta imagen, la cual también será una imagen binaria pero a diferencia de la otra imagen, lo único que será blanco en esta imagen será el contorno del objeto todo lo demás será negro, esta será copiada a un *PictureBox*.

¹² El control *TrackBar* de formularios Windows Forms (también denominado en ocasiones control "deslizante") se utiliza para desplazarse por grandes volúmenes de información o para ajustar visualmente una configuración numérica

Una vez obtenida esta imagen, se realiza otro análisis *pixel a pixel*, de izquierda a derecha y de arriba hacia abajo, en donde obtenemos 4 variables las cuales son: el primer *pixel* de arriba hacia abajo que pertenece al objeto, el pixel que se encuentra más abajo y también pertenece al objeto, los otros dos *pixeles* mas son: el que se encuentra mas a la izquierda y el que se encuentra más a la derecha.

Estas variables nos ayudaran a recortar la imagen original (imagen estática instantánea que se obtuvo en el módulo 1), obteniendo una imagen que contiene el objeto a identificar con el menos color de fondo que sea posible. Esta será enviada al siguiente modulo el cual generara un descriptor.

Un problema que se obtuvo en este modulo es que si una imagen es muy ancha y la fuente de luz no está totalmente perpendicular, genera una sombra que interferirá con la identificación del objeto.

4. Elegir un algoritmo e implementar el módulo que genere un descriptor estocástico¹³ del objeto que se desea identificar.

La generación de este descriptor se llevo a cabo utilizando las componentes *RGB* de cada *pixel*, estas componentes fueron guardadas dentro en un *array*¹⁴, para después obtener la *moda*¹⁵ de cada componente, este descriptor es comparado con los descriptores contenidos en nuestra base de datos y si este se encuentra dentro del margen de error se da por hecho que se trata de este objeto. En este momento se da por concluido el proceso, este proceso se repetirá mientras el usuario no detenga el proceso.

5. Diseñar e implementar la indexación de la base de datos¹⁶ usando los descriptores de los objetos como índices de los objetos a identificar.

La base de datos utilizada contiene el nombre del objeto, así como sus componentes *RGB*, estos datos son guardados en un *HashTable*¹⁷, estos datos son almacenados siguiendo el mismo proceso por el cual se identifica un objeto, solo que al obtener el descriptor este no es comparado sino guardado en el *HashTable*.

Al iniciar por primera vez el programa no contendrá nada en la base de datos así que lo primero que se deberá hacer es almacenar los descriptores de los objetos deseados.

¹³ Métrica que se aplica sobre un sector de la imagen y que nos da una medida de alguna característica estadística de la imagen.

¹⁴ Un array (matriz o vector) es una zona de almacenamiento contiguo, que contiene una serie de elementos del mismo tipo

¹⁵ En Estadística, la moda es el valor con una mayor frecuencia en una distribución de datos.

¹⁶ Conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso

¹⁷ Representa una colección de pares de clave y valor organizados en función del código hash de la clave.

Los descriptores en la base de datos, son guardados y recuperados al ser depositados en un archivo binario.

6. Implementar la unión de los módulos en uno solo por medio de threads.¹⁸

Este fue el mayor problema al realizar este proyecto ya que nunca se logro sincronizar los *threads* de manera efectiva y sin errores, debido a esto se opto por sincronizarlos por medio de un ciclo *while* y variables booleanas.

Existe una variable principal *booleana* la cual es la encargada de iniciar o detener el ciclo *while*, dentro de este ciclo *while* existen las llamadas a los métodos que realizan la tarea de identificar a los objetos, estos métodos tienen variables de retorno *booleanas*, estos métodos al realizar eficazmente su tarea regresan como verdadero, al hacer esto el ciclo *while* puede seguir avanzando hasta lograr la completa realización de todos los métodos, y conllevar a la identificación de los objetos, un mal funcionamiento o la llamada de una *excepción*¹⁹, produce que el programa se detenga totalmente o si la *excepción* es manejada correctamente el programa reinicie la identificación.

¹⁸ Característica del sistema operativo que permite a una aplicación realizar varias tareas concurrentemente.

¹⁹ Es una estructura de control de los lenguajes de programación diseñada para manejar condiciones anormales que pueden ser tratadas por el mismo programa que se desarrolla.

4. Diagrama de Clases.

Este *Figura 1*

Diagrama de Claseses un es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. La imagen que se presenta en este documento tal vez no sea muy explícita por falta de detalle, pero se ha incluido la imagen original "*PT Diagrama de Clases.jpg*", donde se aprecia con mayor detalle el diagrama, asimismo se importo en formato *XML*²⁰, este archivo lleva por nombre "*PT Diagrama de Clases.xml*", se utilizo este formato ya que un archivo *XML*, permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

Para la completa documentación del diagrama de clases se elaboro un documento *CHM*²¹, el cual presenta toda la documentación de las clases atributos y métodos de este trabajo, el nombre del archivo es "*PTDocumentacion.chm*".

Todos los archivos antes mencionados se encuentran en el mismo directorio donde se encuentra este archivo.

²⁰ Extensible Markup Language (lenguaje de marcas extensible), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C).

²¹ Microsoft Compiled HTML Help es un formato privativo de ayuda en línea desarrollado por Microsoft.

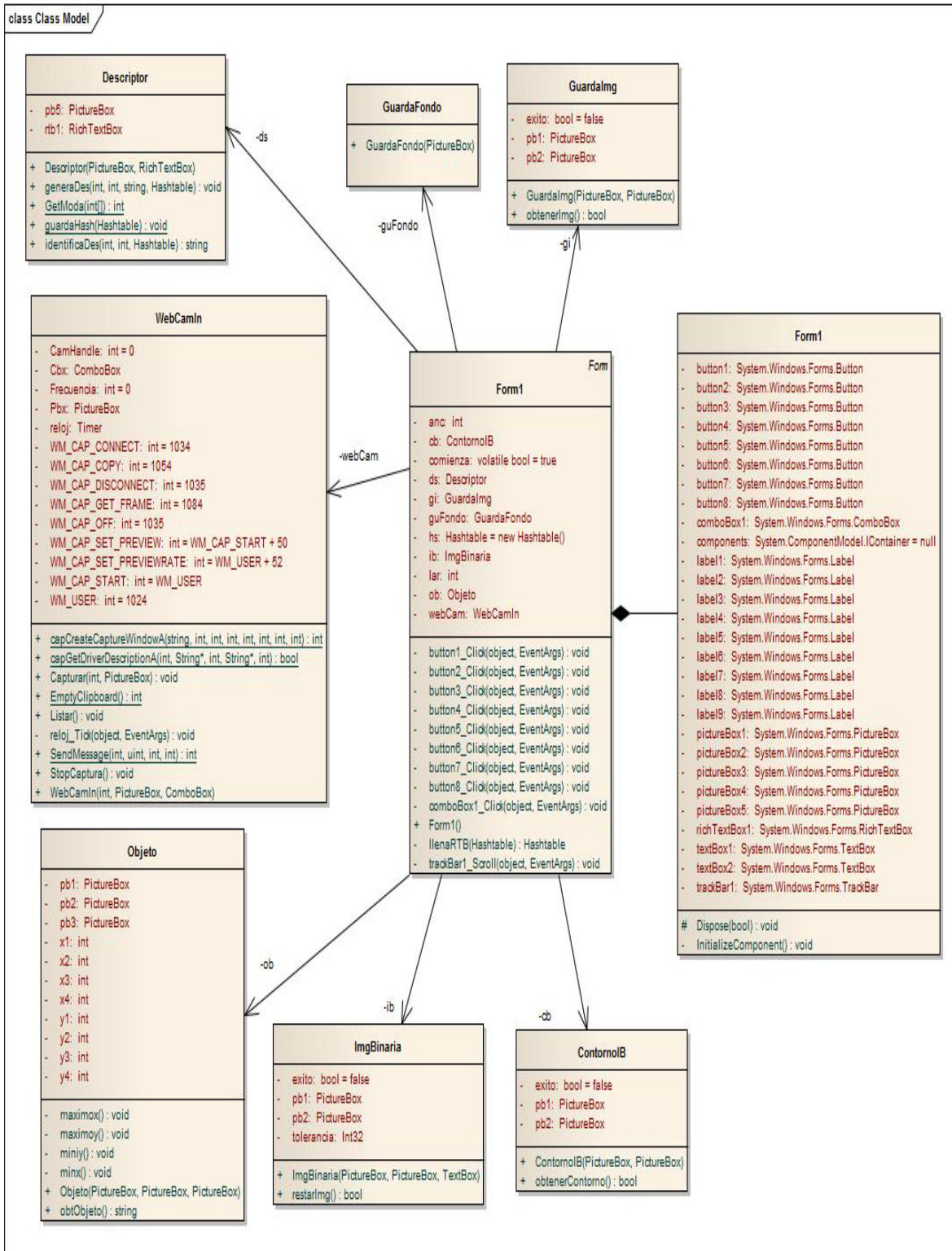


Figura 1

Diagrama de Clases

5. Diagramas de Estados.

Los diagramas de estados, representan gráficamente máquinas de estados finitos. En este reporte utilizaremos tres tipos de diagramas de estado, igualmente que el caso anterior si la imagen pierde detalle se incluyen las imágenes originales, para así poder apreciarlas con pleno detalle.

El *Diagrama de Estados 1* de estados, indica cómo funcionan los módulos entre si y cuál es el orden que se sigue para lograr la identificación de un objeto.

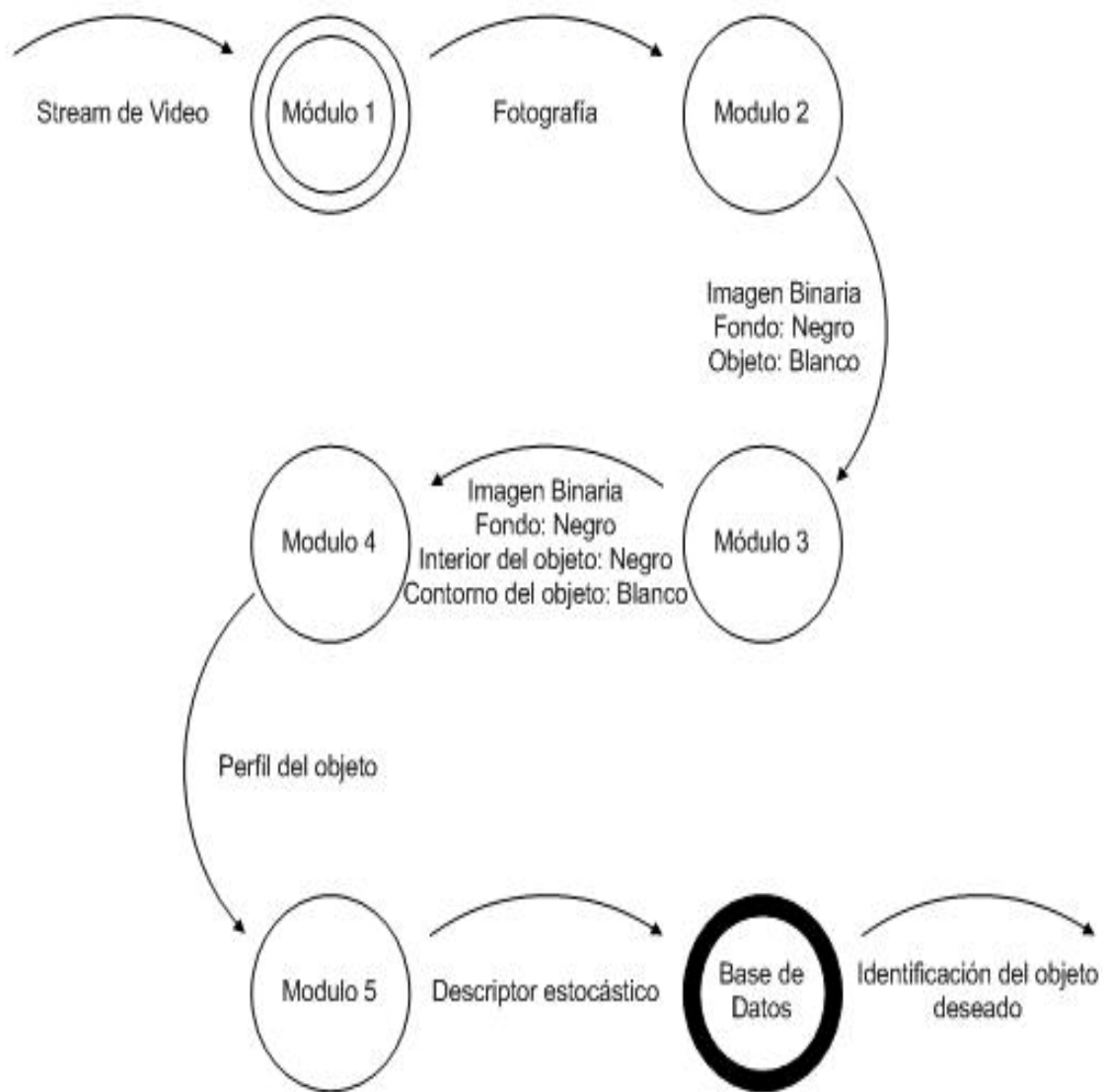


Diagrama de Estados 1

El *Diagrama de Estados 2*, nos proporciona información sobre las entradas y salidas de cada modulo.

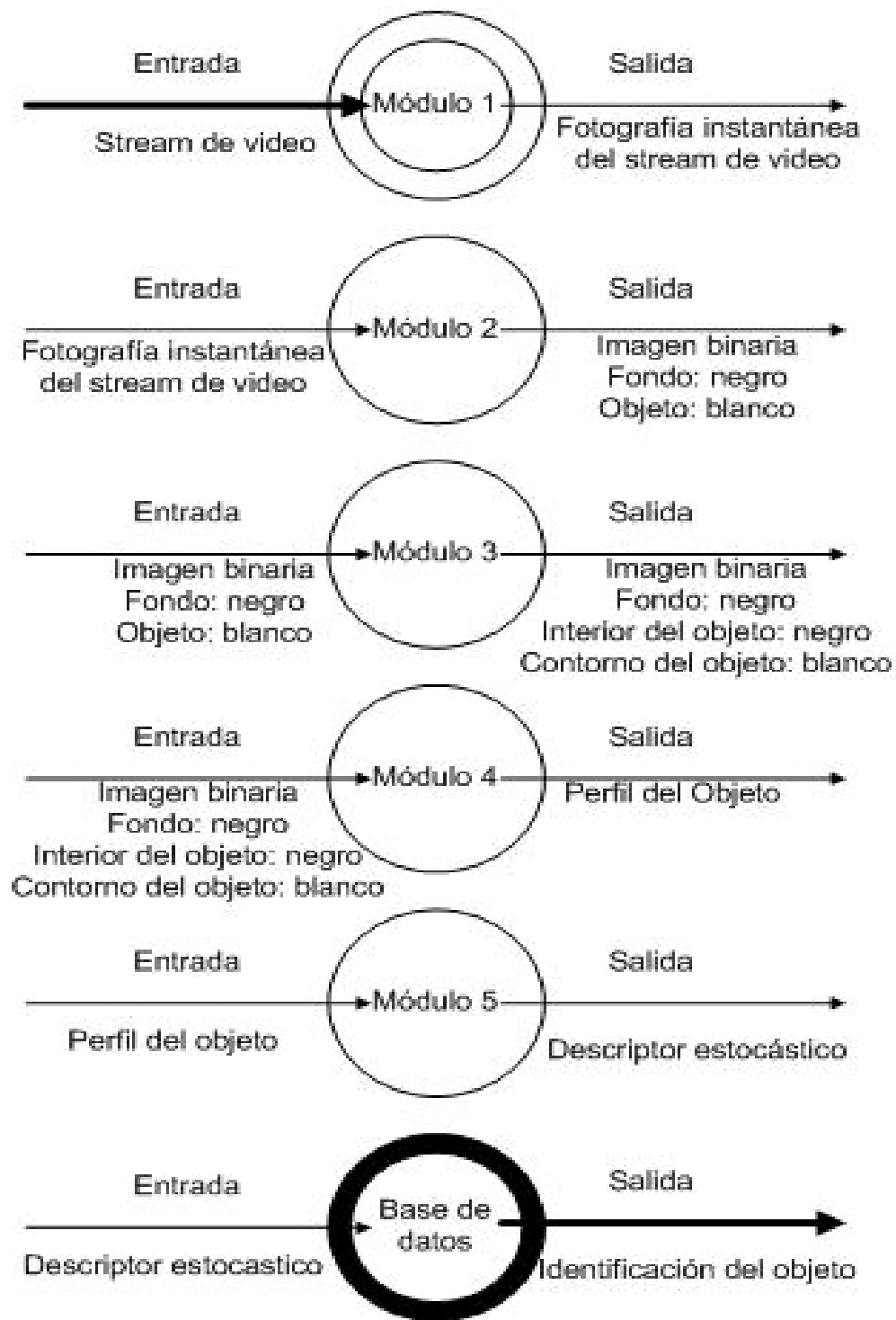


Diagrama de Estados 2

El *Diagrama de Estados 1*, es una forma de ver como se sigue el proceso de identificar a un objeto pero más a nivel de usuario, ya que se hace referencia a como operar el programa y que pasos son lo que se siguen para obtener la identificación del objeto. El contenido de esta imagen también fue exportada al formato *XML* para su futuro uso en cualquier otro sistema, al igual que la imagen original está incluida, para poder analizarla con más detalle.

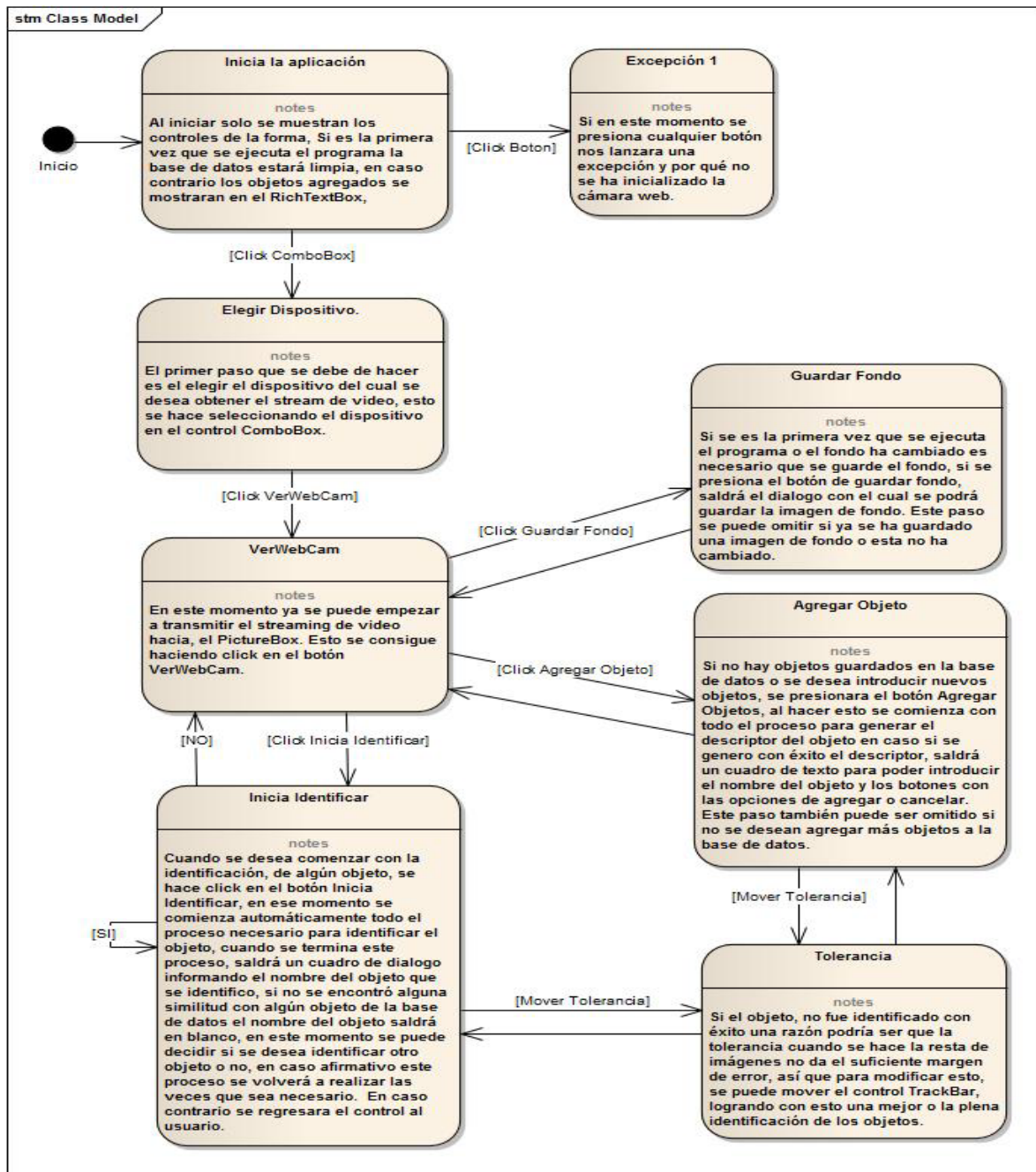


Diagrama de Estados 1

6. Diagrama de Secuencia.

Este *Diagrama de Secuencia* muestra la interacción del usuario, la base de datos, el programa y las clases, a través del tiempo y esta modelado para cada método de la clase. Esta imagen podría no entenderse en este documento pero se adjunta la imagen original para poder apreciarla a detalle, al igual se anexa su formato XML, para su uso en cualquier sistema.

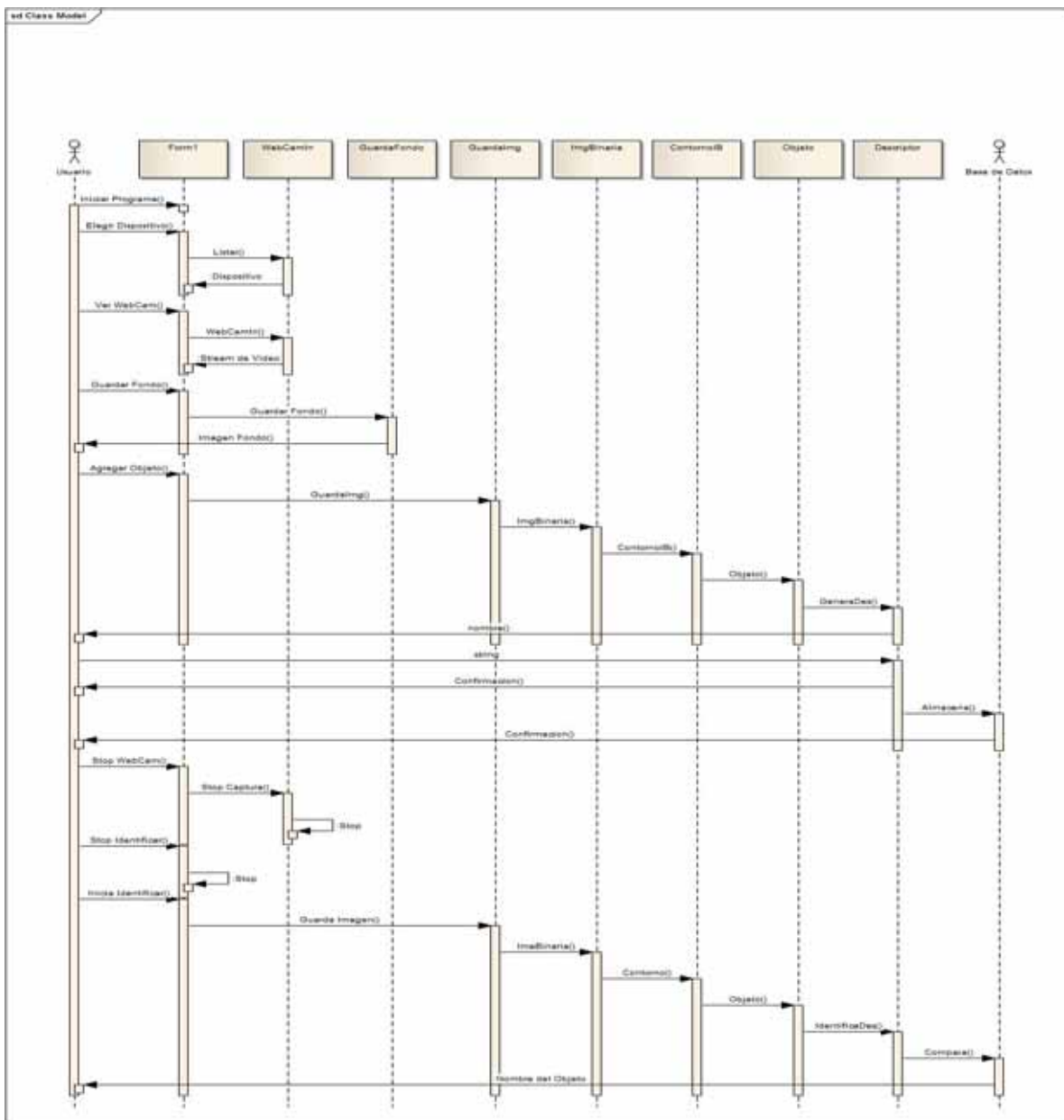


Diagrama de Secuencia

7. Conclusiones.

Este trabajo presenta una implementación que resuelve el problema principal que es la de poder identificar ciertos objetos por medio de manipular digitalmente imágenes.

Inicialmente el proceso de identificación sería para figuras de lego (bloques de plástico interconectables), pero el problema se presentó en que al ser un objeto con demasiada altura, generaba excesiva sombra y esto creó un inconveniente al momento de tratar de eliminar el fondo del objeto que se deseaba identificar, una solución a este problema es, identificar objetos planos que casi no generen sombra o que la fuente de luz esté totalmente perpendicular al objeto de esta forma se eliminaría la sombra y se logra una mejor identificación de los objetos.

Los descriptores que se generaron para los objetos toman pixel por pixel de la imagen y cada pixel se descompone en sus componentes *RGB*, de esta manera por pixel se tienen tres valores los cuales se guardan en tres vectores distintos, uno para cada componente, después se utiliza la moda estadística obteniendo así, tres valores que nos describirían el objeto, en un principio se pensó en lugar de tener tres valores por figura se podría obtener solo uno, esto se obtendría sumando los cuadrados de los tres valores y después sacando la raíz cuadrada, pero este método no funcionó debido a que al generar descriptores de esta manera, los valores obtenidos eran muy similares para objetos muy distintos, entonces se optó por utilizar los tres valores para generar los descriptores.

La sincronización de las clases y métodos se propuso en un principio hacerlo por medio de threads, para que de esta manera se aprovechara más el poder de procesamiento de las computadoras, pero esto no se logró debido a que nunca se pudo establecer una sincronización completamente funcional, debido a esto se optó por un método que no tiene el rendimiento de los threads, sin embargo funcionó, el método utilizado fue usar un ciclo while, con una variable que mientras esta fuera verdadera, los métodos se estarían repitiendo indefinidamente.

La rotación de los objetos presenta cierto problema al ser objetos de color azul ya que si al momento de introducir el objeto en la base de datos, este no se encuentra rotado casi no presenta imagen de fondo, pero al momento en que se desea identificar si se encuentra con cierta rotación generará una imagen que contiene demasiado fondo y esto influye en la identificación de esa figura.

Para finalizar se puede decir que el proyecto funciona y es capaz de identificar ciertas figuras planas y de diferentes colores, con ciertos problemas como los descritos anteriormente, pero en conjunto el programa cumple su finalidad.