

UNIVERSIDAD POLITECNICA SALESIANA

SEDE CUENCA.

FACULTAD DE INGENIERIA

CARRERA: ELECTRONICA

Tesis previa a la obtención del Título de:

Tecnólogo en Electrónica

**“IMPLEMENTACION DE UN PROGRAMA DIDACTICO REALIZADO EN
LABVIEW PARA EL APRENDIZAJE Y DESARROLLO DE LA
MOTRICIDAD DE NIÑOS CON DISFUNCION MOTORA”**

Autor:

Cristian Patricio Valencia Andrade

Director:

Ing. Ítalo Mogrovejo

CUENCA, JUNIO DE 2012

AGRADECIMIENTO

Agradezco a Dios por haberme dado la salud y la inteligencia para terminar con éxitos esta etapa de mis estudios.

A la Universidad Politécnica Salesiana por brindarme los conocimientos necesarios para poder desenvolverme en mi vida profesional con un alto nivel académico y continuar mis estudios.

Al Ing. Ítalo Mogrovejo por guiarme en la elaboración de mi proyecto de tesis.

EL AUTOR

DEDICATORIA

A mi madre, quien sin su apoyo incondicional,
dedicación y esfuerzo no hubiese sido posible
culminar esta etapa de mis estudios.

RESUMEN

El presente trabajo se enmarca en los campos de visión artificial y el procesamiento de imágenes, haciendo uso de los conocimientos adquiridos durante las cátedras de “Procesamiento Digital de Señales “ y “Sensores - Transductores”, se ha desarrollado la “IMPLEMENTACION DE UN PROGRAMA DIDACTICO REALIZADO EN LABVIEW PARA EL APRENDIZAJE Y DESARROLLO DE LA MOTRICIDAD DE NIÑOS CON DISFUNCION MOTORA”

El sistema se basa en que el participante comanda un programa didáctico, con ejercicios que vayan de acuerdo a su capacidad intelectual, únicamente con el movimiento de brazos.

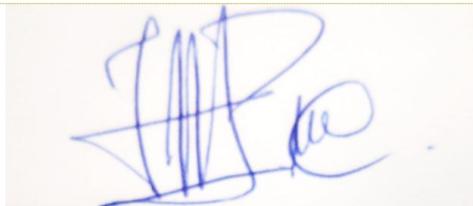
El programa puede ser comandado hasta una distancia de 6m sin ningún tipo de cable, con la ayuda de una aplicación en LabVIEW 10, que permite el reconocimiento de patrones de colores por medio de una cámara USB.

Ing. Ítalo Mogrovejo

CERTIFICA

Haber dirigido y revisado prolijamente cada uno de los capítulos de este trabajo de tesis, realizado por el señor Cristian Patricio Valencia Andrade.

Por cumplir los requisitos, autorizo su presentación.

A handwritten signature in blue ink, appearing to be 'Ítalo Mogrovejo', written on a light-colored background. The signature is somewhat stylized and includes a small flourish at the end.

Ing. Ítalo Mogrovejo

DIRECTOR DE TESIS

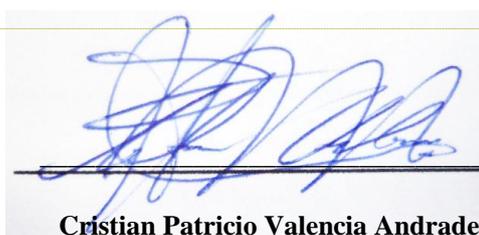
Con formato: Fuente:
(Predeterminado) Times New Roman
12 pto, Negrita

DECLARATORIA DE RESPONSABILIDAD

Yo, Cristian Patricio Valencia Andrade, autor del presente Trabajo de Tesis titulado, “IMPLEMENTACION DE UN PROGRAMA DIDACTICO REALIZADO EN LABVIEW PARA EL APRENDIZAJE Y DESARROLLO DE LA MOTRICIDAD DE NIÑOS CON DISFUNCION MOTORA” declaro que:

Los conceptos desarrollados, análisis realizados y las conclusiones del presente trabajo, son de exclusiva responsabilidad del autor.

Cuenca, junio de 2012



Cristian Patricio Valencia Andrade

Con formato: Fuente:
(Predeterminado) Times New Roman
12 pto, Negrita

**LOS MIEMBROS DEL TRIBUNAL DE CALIFICACION Y
SUSTENTACION DE LA TESIS**

**“IMPLEMENTACION DE UN PROGRAMA DIDACTICO REALIZADO EN
LABVIEW PARA EL APRENDIZAJE Y DESARROLLO DE LA
MOTRICIDAD DE NIÑOS CON DISFUNCION MOTORA”**

Que la presente tesis ha sido debidamente revisada y por lo tanto

APROBADA

Presidente Del Tribunal

Miembro Del Tribunal

Miembro Del Tribunal

Cuenca, junio de 2012

ÍNDICE GENERAL

Introducción.....	4
Antecedentes.....	4
Capítulo 1. Procesamiento de imágenes y visión por computadora con NI-IMAQ.5	
1.1 Entorno IMAQ VISION en LabVIEW.....	5
1.1.1 ¿Qué es LabVIEW?	5
1.1.2 Entorno.....	5
1.1.3 NI VISION.....	14
1.1.4 VISION AND MOTION.....	15
1.1.5 VISION UTILITIES.....	15
1.1.6 MACHINE VISION.....	16
1.1.7 Administración del buffer de la memoria.....	16
1.1.7.1 IMAQ CREATE.....	17
1.1.7.2 IMAQ DISPOSE.....	19
1.2 Imágenes digitales.....	19
1.2.1 Definición.....	19
1.2.2 Propiedades de una imagen digital.....	21
1.2.3 Tipos de imágenes.....	22
1.2.3.1 Imágenes en escala de grises.....	23
1.2.3.2 Imágenes a color.....	23
1.2.3.3 Imágenes complejas.....	24
1.2.4 Archivos de imagen.....	25
1.2.4.1 Imágenes de mapa de bits.....	25

1.2.4.2	Imágenes vectoriales.....	26
1.2.4.2	Tipos de archivo	26
1.3	Adquisición de imágenes con NI-IMAQ para cámaras USB.....	28
1.3.1	NI_IMAQ para cámaras USB.....	28
1.4	Procesamiento y análisis de imágenes.....	31
1.4.1	Procesamiento de imágenes.....	31
1.4.2	Análisis de imágenes.....	32
Capítulo 2.	Interacción Electromecánica entre el dedal y el software.....	35
2.1	Descripción de la funcionalidad del dedal y la luz.....	35
2.2	Selección del ambiente controlado	35
2.3	Descripción de la funcionalidad de la cámara USB.....	40
2.3.1	Cámara Digital y Analógica.....	41
2.3.2	Tipo de salida de Video	41
2.3.2.1	Formato de video.....	42
2.3.2.2	Resolución.....	42
2.3.2.3	Características del sensor de cámaras digitales modernas.	42
2.3.2.3.1	Sensor CCD.....	43
2.3.2.3.2	Sensor CMOS.....	44
2.4	Selección de la cámara para la aplicación.....	45
2.5	Montaje de la cámara.....	46
Capítulo 3.	Diseño e implementación.....	47
3.1	Determinación de parámetros de diseño.....	47

3.2	Diseño del patrón a reconocer.....	50
3.2.1	Creación de la Plantilla.....	51
3.2.2	Selección de Plantillas	52
3.3	Diseño del programa en LabVIEW para el seguimiento de luz.....	52
3.3.1	Pestaña de Presentación.....	54
3.3.2	EL botón “CONTINUAR”.....	53
3.3.2.1	Adquisición.....	53
3.3.2.2	Detección y movimiento.....	56
3.3.3	Pestaña de Calibración.....	57
3.3.4	Pestaña de Selección de menú.....	58
3.4	Diseño y programación en LabVIEW de juegos para niños.....	60
3.4.1	Desarrollo del juego “ENTRENADOR”.....	61
3.5	Desarrollo de juegos en Adobe Flash Professional.....	65
3.5.1	Adobe Flash Professional.....	66
3.5.2	Procedimiento para el desarrollo de juegos.....	68
Capitulo 4. Pruebas realizadas y publicación de resultados.....		71
4.1	Identificar el color luz apropiado.....	71
4.1.1	Datos obtenidos en base a prueba y error.....	72
4.2	Manejo del sistema.....	72
4.2.1	Datos obtenidos del manejo del sistema.....	73
4.3	Análisis de resultados.....	74
Conclusiones y recomendaciones.....		76
Anexos.....		81

Introducción

El presente trabajo describe el desarrollo del diseño e implementación de un sistema en LabVIEW y su módulo de procesamiento de imágenes para el aprendizaje y desarrollo de la motricidad de niños con problemas de psicomotricidad.

Antecedentes

Desde hace pocos años, han surgido consolas de juegos con mando inalámbrico tales como Wii Motion Plus para consolas Wi, Play Station Move para PlayStation 3 y Kinect de Xbox 360; sin embargo todos estos dispositivos no son de fácil acceso a causa de sus elevados costos y muchos de ellos no contienen juegos con fines educacionales. En la actualidad existen un gran porcentaje de niños con déficit de atención y problemas de motricidad por lo que resulta dificultoso para los tutores iniciar un proceso de aprendizaje rápido y eficaz.

CAPÍTULO 1. PROCESAMIENTO DE IMÁGENES Y VISIÓN POR COMPUTADORA CON NI-IMAQ

1.1 ENTORNO IMAQ VISION EN LABVIEW

1.1.1 ¿QUÉ ES LABVIEW?

LabVIEW es el acrónimo de Laboratory Virtual Instrument Engineering Workbench.

Es un lenguaje de programación gráfica en el que se pueden crear aplicaciones de una forma rápida y sencilla, su logo se muestra en la figura 2-1.



Figura 2-1 Logo de LabVIEW

FUENTE: LabVIEW (modificado el Autor)

LabVIEW posee un ambiente de desarrollo gráfico dinámico, con funciones integradas para realizar adquisición de datos, control de instrumentos, análisis de mediciones, presentaciones de datos entre otras.

A diferencia de los lenguajes de propósito general, LabVIEW provee funcionalidad específica para que pueda acelerar el desarrollo de aplicaciones de medición, control y automatización.

1.1.2 ENTORNO

El usuario interactúa a través de dos pantallas principales: El Panel frontal y el diagrama de bloques.

El **panel frontal** es la parte que ve el usuario, suele tener un fondo gris. Como se aprecia en la figura 1.2, contiene entradas y salidas. Los datos pueden ser introducidos a través del teclado o el mouse gracias al uso de controles, interrupciones, etc. y los resultados mostrados en la pantalla a través de gráficos, leds, tanques, indicadores, entre otros.

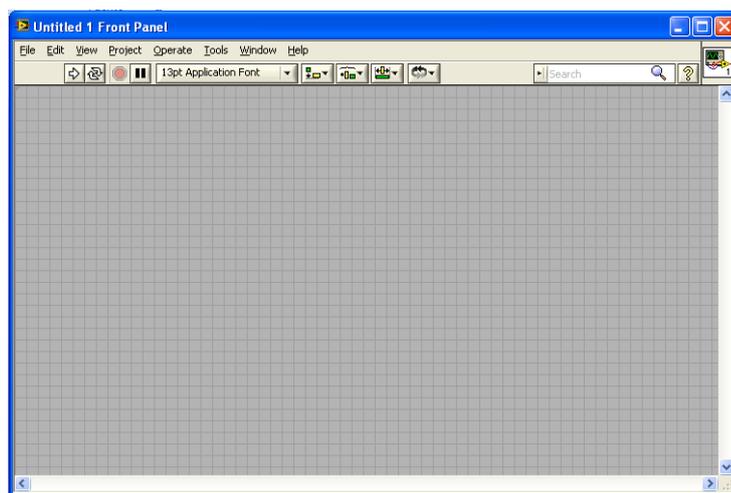


Figura 1.2 Presentación Panel frontal

FUENTE: LabVIEW (modificado el Autor)

La ventana del **diagrama de bloques** es donde se realiza la programación y suele tener fondo blanco, aquí se ubican los objetos que en conjunto componen la programación grafica que emplea LabVIEW, dejando atrás la programación tradicional que utiliza líneas de texto de código como se aprecia en la figura 1.3.

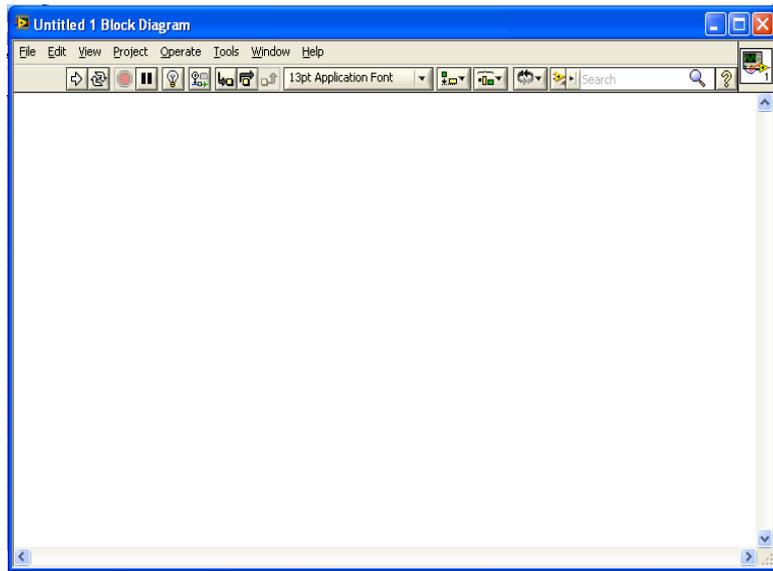


Figura 1.3 Presentación Diagrama de Bloques

FUENTE: LabVIEW (modificado el Autor)

En la parte superior de estas ventanas se sitúa una barra con varias herramientas. En el Diagrama de bloques esta barra tiene algunas opciones como se aprecia en la figura 1.4.

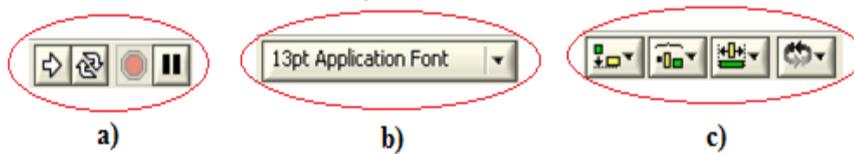


Figura 1.4 Paleta principal del diagrama de bloques

FUENTE: LabVIEW (modificado el Autor)

El primer grupo de herramientas sirve para controlar la ejecución de un programa en LabVIEW. El segundo grupo ayuda a la depuración. El tercer grupo se utiliza para

alinear, distribuir, controlar el tamaño, agrupar y ordenar objetos con el fin de llevar un orden y mejorar la visualización.

En el lateral derecho tanto del Panel Frontal como del Diagrama de Bloques aparece el icono que representa el VI, al hacer doble clic sobre este permite la modificación de apariencia.

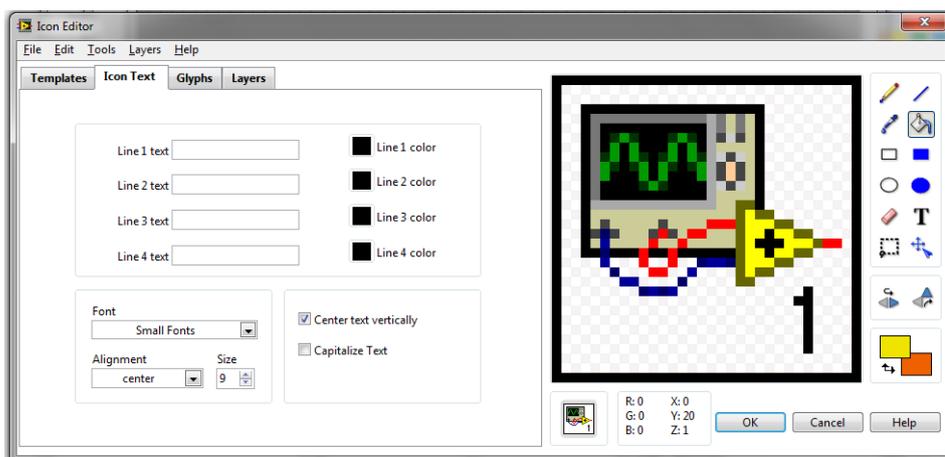


Figura 1.5 a) Ventana para personalizar el icono del VI

FUENTE: LabVIEW (modificado el Autor)

También existe una paleta muy útil y necesaria llamada Tools Palette como se aprecia en la figura 1.6. Esta paleta permite crear, modificar, depurar Vis utilizando sus diferentes opciones. Si la paleta de herramientas no aparece se puede activar seleccionando cualquiera de los paneles en: Windows → Show Tools Palette.



Figura 1.6 Paleta de herramientas

FUENTE: LabVIEW (modificado el Autor)

A continuación se detallan las utilidades de cada uno de los iconos que aparecen dentro de la paleta de herramienta (Tools Palette).

Herramienta de Selección Automática. Al estar habilitado esta herramienta, cuando se mueve el cursor sobre los diferentes objetos en el panel frontal o diagrama de bloques, LabVIEW selecciona automáticamente la herramienta correspondiente de la paleta. En resumen cada icono de la paleta cambia el comportamiento del cursor en LabVIEW, con lo que se puede posicionar, operar y editar las diferentes tareas de los Vis.



Figura 1.7 Herramienta de selección Automática

FUENTE: LabVIEW (modificado el Autor)

Herramienta de Operación. Cuando el cursor del mouse cambia al icono que se muestra en la figura 1.8, permite cambiar los valores de un objeto de control en el panel frontal, o en el diagrama de bloques para cambiar el valor de un booleano.



Figura 1.8 Herramienta de operación

FUENTE: LabVIEW (modificado el Autor)

Herramientas de posicionamiento. Cuando el cursor del mouse cambia al icono que se muestra en la figura 1.9 permite seleccionar, mover o dar nuevas dimensiones a los distintos objetos.



Figura 1.9 Herramienta de posicionamiento

FUENTE: LabVIEW (modificado el Autor)

Herramienta de Etiquetado. Cuando el cursor del mouse cambia al icono que se muestra en la figura 1.10, permite modificar el texto en un control, crear etiquetas, y cambiar las propiedades de los textos.



Figura 1.10 Herramienta de Etiquetado

FUENTE: LabVIEW (modificado el Autor)

Herramienta de cableado. Cuando el cursor del mouse cambia al icono que se muestra en la figura 1.11, permite realizar las conexiones de los objetos en el diagrama de bloques.



Figura 1.11 Herramienta de cableado

FUENTE: LabVIEW (modificado el Autor)

Herramienta de menú. Despliega un menú con diferentes opciones en cada uno de los objetos.



Figura 1.12 Herramienta de menú

FUENTE: LabVIEW (modificado el Autor)

Herramienta de desplazamiento. Permite desplazarse a través de una ventana sin utilizar las barras de desplazamiento.



Figura 1.13 Herramienta de desplazamiento

FUENTE: LabVIEW (modificado el Autor)

Herramienta de puntos de detención o de quiebre. Permite establecer “puntos de parada” en una aplicación, de tal manera que la aplicación termine cuando se llegue allí.



Figura 1.14 Herramienta de puntos de detención

FUENTE: LabVIEW (modificado el Autor)

Herramienta de pruebas. Permite colocar puntos de prueba intermedios en una aplicación. Los mismos que permiten observar la información de los valores calculados.



Figura 1.15 Herramienta de pruebas

FUENTE: LabVIEW (modificado el Autor)

Herramienta de copiado de color. Copia colores que aparecen en la ventana activa para ser usado en otros sitios.



Figura 1.16 Herramienta de copiado de color

FUENTE: LabVIEW (modificado el Autor)

Herramienta de color. Permite cambiar el color de los objetos, hacerlos transparentes, y dar una apariencia adecuada en el panel frontal.



Figura 1.17 Herramienta de color

FUENTE: LabVIEW (modificado el Autor)

Paleta de funciones. Están todos los objetos que se pueden utilizar en la implementación del programa del VI, estas pueden ser funciones aritméticas, señales de entrada /salida de datos a fichero, adquisición de señales, temporización de la ejecución del programa, entre otras.

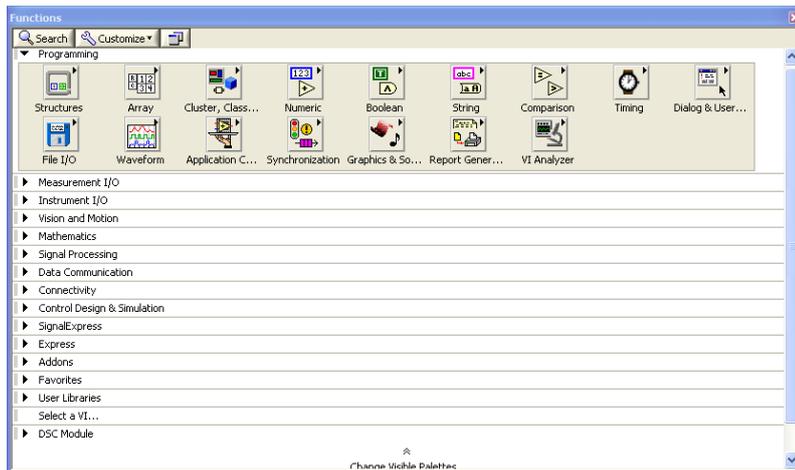


Figura 1.18 Paleta de Funciones

FUENTE: LabVIEW (modificado el Autor)

Paleta de controles/indicadores. Se muestra en la figura 1.19 y se utiliza únicamente en el panel frontal. Contiene todos los controles e indicadores que se emplearan para crear la interfaz del VI con el usuario

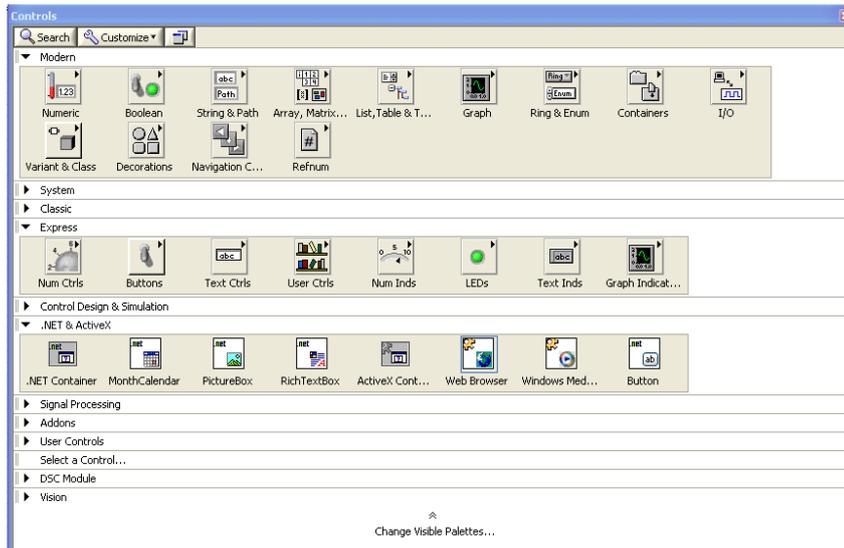


Figura 1.19 Paleta de Controles

FUENTE: LabVIEW (modificado el Autor)

1.1.3 NI VISION

Bastante útil para el análisis y el procesamiento de imágenes, contiene una librería con más de 400 Vis.

Posee una biblioteca de poderosas funciones para el procesamiento de visión y el Vision Assistant, un entorno interactivo para desarrolladores que necesitan generar rápidamente prototipos para aplicaciones o que necesitan inspección fuera de línea

IMAQ VISION de LabVIEW requiere usar las librerías de NI-IMAQ para Cámaras USB las mismas que se explican en líneas posteriores en este capítulo.

1.1.4 VISION AND MOTION

Proporciona paquetes que posibilitan la combinación de la visión artificial y la tecnología de control de movimiento, a continuación en la figura 1.20 se muestra el menú de Visión y Movimiento.



Figura 2.20 Menú de funciones Vision And Motion

FUENTE: LabVIEW (modificado el Autor)

1.1.5 VISION UTILITIES.

Permite crear y manipular imágenes extraídas desde archivos, o directamente de una webcam, establecer las regiones de interés, calibración de imágenes, entre otras; en la figura 1.21 se muestra el menú que proporciona este paquete.

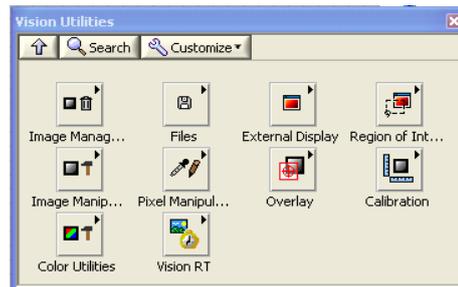


Figura 1.21 Menú Vision Utilities

FUENTE: LabVIEW (modificado el Autor)

1.1.6 MACHINE VISION

Permite realizar trabajos de inspección de visión de máquina comunes, incluyendo la presencia o la ausencia de partes en una imagen, medir dimensiones, localización de bordes, busca de patrones, selección de regiones, clasificación, etc.

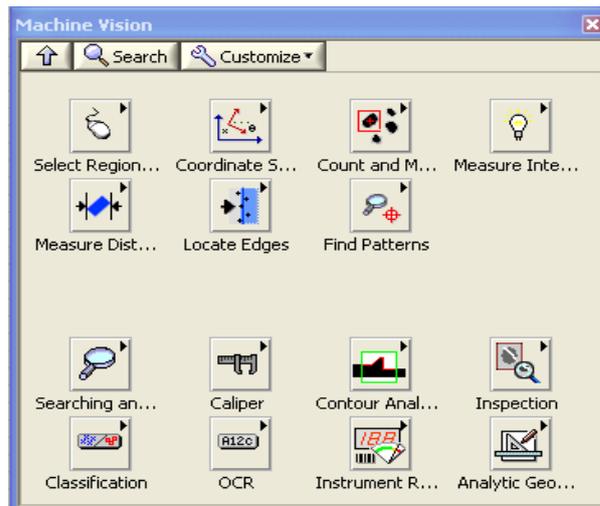


Figura 1.22 Menú Machine Vision

FUENTE: LabVIEW (modificado el Autor)

1.1.7 ADMINISTRACION DEL BUFFER DE LA MEMORIA

En el menú de la ventana Vision Utilities, está la paleta de funciones de Image Managment, sirve para crear, eliminar, adquirir, leer, modificar atributos de una imagen y crear copias de las mismas.

La figura 1.23 muestra las diferentes funciones necesarias para la administración de la imagen.

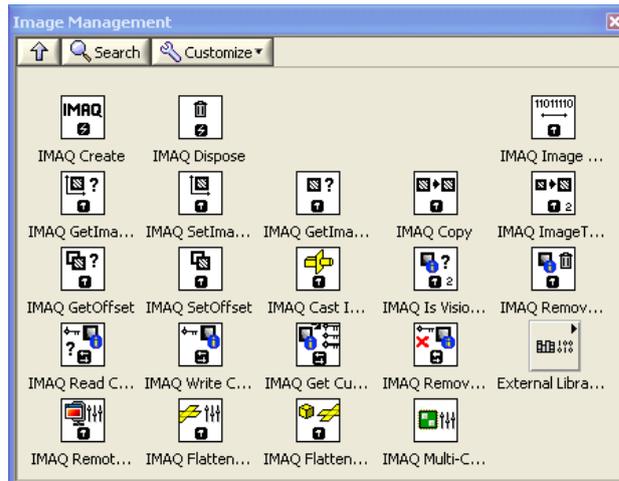


Figura 1.23 Menú Image Managment

FUENTE: LabVIEW (modificado el Autor)

1.1.7.1 IMAQ CREATE

Se muestra en la figura 1.24, permite crear una posición de memoria temporal para una imagen. Este Vi se utiliza siempre con el IMAQ DISPOSE.

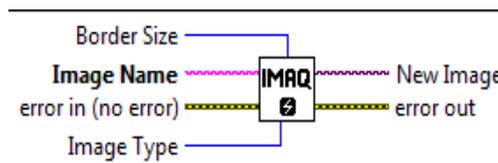


Figura 1.24 IMAQ CREATE

FUENTE: LabVIEW (modificado el Autor)

Dónde:

El Tamaño de Borde (Border Size): define el tamaño, en píxeles, del borde a crear alrededor de una imagen. Se determina desde el inicio si una imagen va a ser procesada posteriormente usando las funciones que requieren uso de bordes; los bordes de una imagen nunca son mostrados o almacenados en un archivo. En la figura 1.25 se muestra dos imágenes de 8x6 píxeles, en la primera con borde 0 y la segunda con borde 2.

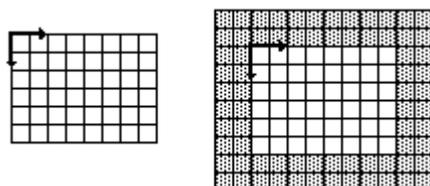


Figura 2-26 Borde de una imagen

FUENTE: Ayuda de LabVIEW (modificado el Autor)

El Nombre de la imagen (Image Name): es el nombre que tiene la imagen creada. Cada imagen tener un nombre que no se repita.

Error de entrada (error in): especifica el estado de error antes de ejecutar el VI o función.

El Tipo de Imagen (Image Type): establece el tipo de imagen a procesar, puede ser: escala de grises, compleja, de color, etc.

Imagen nueva (New Image): es la salida de la imagen que es utilizada como referencia para otras funciones de NI Visión. Múltiples imágenes pueden ser creadas en una aplicación de LabVIEW.

1.1.7.2 IMAQ DISPOSE

Destruye la imagen creada con el IMAQ CREATE y libera el espacio de memoria RAM ocupado por esta. Varias imágenes distintas pueden ser utilizadas. Este Vi se visualiza en la figura 1.26

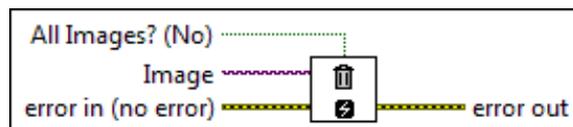


Figura 1.26 IMAQ DISPOSE

FUENTE: LabVIEW (modificado el Autor)

Todas las imágenes (All Images): determina si se elimina una imagen o todas las asociadas previamente con este VI dependiendo si es falsa o verdadera.

Imagen (Image): Especifica la referencia o nombre de la imagen a destruir.

Error de salida (error Out): genera información del error después de ejecutar el VI.

1.2 LA IMAGEN DIGITAL

1.2.1 DEFINICION

Una Imagen digital es un arreglo en dos dimensiones de valores que representan la intensidad luminosa.

Una imagen está en función de la intensidad luminosa $f(x,y)$, donde f representa la intensidad en el punto (x,y) , donde x e y representan las coordenadas espaciales de la unidad básica de la imagen, o pixel.

El punto $(0,0)$ es una referencia para las coordenadas de una imagen y generalmente se ubica en la parte superior izquierda como se observa en la figura 1.27

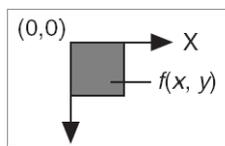


Figura 1-27 Referencia espacial de un pixel en el punto $(0,0)$

FUENTE: (El Autor)

En el procesamiento de una imagen digital, un sensor convierte una imagen en un número finito de píxeles. El sensor asigna a cada píxel una coordenada y un nivel de gris o color que especifique el brillo o el color del píxel, como se indica en la figura 1.28

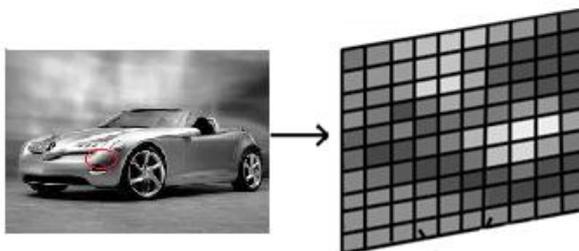


Figura 1.28 Píxeles extraídos de una imagen

FUENTE: (El Autor)

1.2.2 PROPIEDADES DE UNA IMAGEN DIGITAL

Una imagen digital tiene tres propiedades básicas: resolución, definición y número de planos.

Resolución: Se refiere al número de filas y columnas que forman los píxeles de una imagen, en DSP generalmente a las filas se les denota con la letra m y a las columnas con la n .

Definición: Indica el número de tonos que se puede ver en la imagen. La profundidad de bits de una imagen es el número de bits utilizados para codificar el valor de un píxel. Para una representación en n bits, la imagen tiene una definición de 2^n , lo cual significa que un píxel puede tener 2^n valores diferentes. Por ejemplo, si n es igual a 8 bits, un píxel puede tener 256 valores diferentes como se observa en la Figura 1.29, que van desde 0 hasta 255. Si n es igual a 16 bits, un píxel puede tener 65536 valores diferentes que van desde 0 hasta 65535 o desde -32768 hasta 32767. Actualmente, IMAQ Vision sólo soporta un rango de -32768 a 32767 para imágenes de 16 bits.

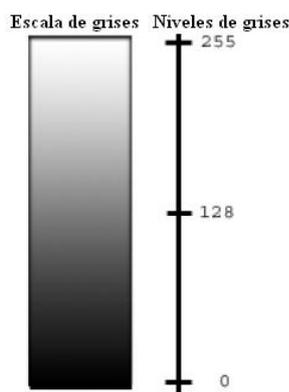


Figura 1.29 256 diferentes niveles de gris

FUENTE: (El Autor)

IMAQ VISION puede procesar imágenes con 8 bits, 10 bits, 12 bits, 14 bits, 16 bits, punto flotante, codificación de colores. La manera en que se codifica la imagen depende de la naturaleza del dispositivo de adquisición de imágenes, el tipo de procesamiento de imágenes que se necesita usar, y el tipo de análisis que necesita realizar. Por ejemplo, la codificación de 8 bits es suficiente si necesita obtener la información acerca del contorno de los objetos que se encuentran en una imagen. Sin embargo, si la aplicación hace uso de reconocimiento de colores, debe usar imágenes digitales a color.

Número de Planos: corresponde al número de matrices de los píxeles que componen la imagen. Una imagen a escala de grises o pseudo-color se compone de un solo plano. Una imagen de color se compone de tres planos: uno para el componente rojo, componente azul, y componente verde, es así que una imagen en color es la combinación de los tres arreglos de píxeles correspondientes a los colores rojo, verde y azul en una imagen RGB, esto se verá con mayor profundidad en las secciones siguientes. En imágenes HSL se definen por los valores correspondientes a tono, saturación y luminosidad, estas últimas no son objeto de estudio para el desarrollo de esta tesis por lo que no se hará énfasis en ello.

1.2.3 TIPOS DE IMÁGENES

Las librerías de IMAQ Vision pueden manipular tres tipos de imágenes: en escala de grises, de color, y las imágenes complejas. Aunque IMAQ Vision maneja los tres tipos de imágenes, algunas operaciones no son posibles en ciertos tipos de imágenes. En la misma resolución espacial, una imagen de color ocupa cuatro veces el espacio de memoria de una imagen de 8 bits en escala de grises, y una imagen compleja ocupa ocho veces la memoria de la misma imagen de 8 bits en escala de grises.

1.2.3.1 IMÁGENES EN ESCALA DE GRISES

Una imagen en escala de grises se compone de un solo plano de píxeles.

Cada pixel puede ser codificado mediante uno de los siguientes formatos numéricos:

- Enteros de 8 bits sin signo que representan los valores de escala de grises entre 0 y 255.
- Enteros de 16 bits con signo que representan los valores de escala de grises entre -32768 y +32767.
- Un solo número de punto flotante de precisión, codificado usando cuatro bytes, que representa los valores en escala de grises que van desde $-\infty$ a ∞ .

1.2.3.2 IMÁGENES A COLOR

Toda imagen a color se encuentra codificada como una imagen en RGB, HSL, HSV o CMYK.

El modelo CMY (cian, magenta, yellow) se basa en una representación del color por un método sustractivo. Contrariamente a lo que sucede con el RGB, este modelo parte de la presencia total de luz (blanco) y va sustrayendo colores para formar todo el espectro (recordemos que el blanco es la concurrencia de todos los colores y el negro es la ausencia de los mismos).

Las imágenes de color RGB almacenan la información en los planos rojo, verde y azul. En imágenes de color HSL, la información se encuentra en los planos de tono, saturación y luminancia. En ambos modelos, cada plano es de 8 bits y tienen 8 bits adicionales que no son usados. Esta representación se conoce como codificación en 32 bits y es la que maneja generalmente las cámaras USB estándares.

El modelo RGB parte del negro (ausencia de color) y los colores se suman hasta llegar a su valor máximo 255, 255,255 que es el blanco.

LabVIEW también maneja otros muchos formatos dependiendo de la necesidad o la cantidad de información que se requiera extraer de una imagen, una de ellas es el RGB U64 la información de color reside en planos de 16 bits para cada plano, un campo adicional de 16 bits que no es usado.

En la figura 1.30 se muestra como se complementan el formato de imagen RGB y el CMYK para formar diversos colores

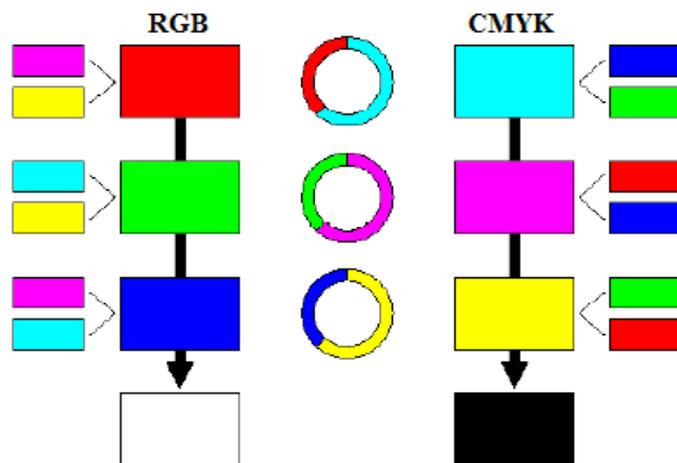


Figura 1.30 Modelo del formato de imagen RGB y el CMYK

FUENTE: (El Autor)

1.2.3.3 IMÁGENES COMPLEJAS

Una imagen compleja contiene información de la frecuencia de una imagen en escala de grises. Se puede crear una imagen compleja mediante la aplicación de una transformada Rápida de Fourier (FFT) a una imagen en escala de grises, a fin de poder realizar operaciones en dominio de la frecuencia sobre la imagen.

Cada píxel en una imagen compleja es codificada como dos valores de punto flotante, que representan a los componentes real e imaginario del píxel complejo. Se pueden extraer cuatro valores de una imagen compleja: la parte real, parte imaginaria, la magnitud y fase.

1.2.4 ARCHIVOS DE IMÁGENES

Existen dos modos principales para manipular la información que integra una imagen digital. Estos modos son las imágenes de mapa de bits y las imágenes vectoriales.

1.2.4.1 IMÁGENES DE MAPA DE BITS

Están formadas por un conjunto de celdas llamadas píxeles, se les asigna un color y luminancia propios, es por esta razón que un píxel se define como la unidad de información. Un píxel no depende del tamaño ya que puede ser tan pequeño como 0.1 mm o muy grande 1cm.

Cuando creamos una imagen de mapa de bits se genera una rejilla específica de píxeles. Por esto, al modificar su tamaño, transformamos, a su vez, la distribución y coloración de los píxeles, por lo que los objetos, dentro de la imagen, suelen deformarse. Esto es porque los objetos pierden o ganan algunos de los píxeles que los definen. Gracias a esta característica, que siempre hay que tener en cuenta, las imágenes de mapa de bits se crean con un tamaño determinado y pierden calidad si se modifican sus dimensiones,

estas características permiten que estas imágenes se obtengan desde una cámara digital. [13]

1.2.4.2 IMÁGENES VECTORIALES

Son de menor tamaño en comparación con los mapas de bits, porque organizan la información de forma simple, esta simplicidad radica en generar los objetos que conforman una imagen a través de trazos geométricos determinados por cálculos y fórmulas matemáticas. De manera tal que los gráficos vectoriales se visualizan a partir de las coordenadas de una línea guardadas como referencia, mismas que forman los objetos a partir de la definición matemática de los puntos y líneas rectas o curvas, permitiendo manipular un objeto dentro de la imagen sin que se alteren las demás.

1.2.4.2 TIPOS DE ARCHIVO

BMP: Este es el formato de las imágenes de mapa de bits que utiliza Windows, los archivos son muy grandes dado la escasa compresión que alcanzan.

PSD: Este es el formato de Adobe Photoshop y, por lo mismo, es el único que admite todas las funciones que este programa contiene. Sin embargo, su uso se centra en la manipulación de la imagen y no tanto para ser empleado en publicaciones digitales.

TIF: se utiliza para imágenes de mapa de bits y es admitido prácticamente por todas las aplicaciones de autoedición y tratamiento de imágenes

GIF: Se emplea para las imágenes de tonos no continuos o cuando hay grandes áreas de un mismo color ya que utiliza una paleta de color indexado que puede tener un máximo de 256 colores.

JPG O JPEG: Se utiliza usualmente para almacenar fotografías y otras imágenes de tono continuo. Gracias a que utiliza un sistema de compresión que de forma eficiente reduce el tamaño de los archivos.

PNG: fue creado para su distribución en la red posee gran parte de las ventajas de un GIF y de un JPG. Por ejemplo, permite altos niveles de compresión, además, permite utilizar la técnica de la indexación para crear colores transparentes, o transparencias degradadas.

En LabVIEW el bloque encargado de localizar y leer la imagen del usuario es el IMAQ READFILE, se muestra en figura 1.31.

Esta sub-función se encuentra en *Functions>Vision and Motion>Vision utilities>Files*, permite la lectura de imágenes en formatos JPEG, BMP ,PNG, TIFF O AIPD, convierte la lectura de pixeles en la imagen antes definida en el IMAQ CREATE.

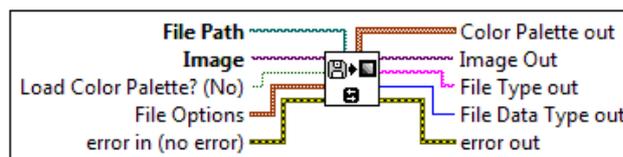


Figura 1.31 IMAQ READFILE

FUENTE: Ayuda de LabVIEW (modificado el Autor)

Image: es una referencia a la imagen a ser leída.

Image Out: es una referencia a la imagen de destino.

File Type Out: indica el tipo de archivo que se lee. Esta cadena devuelve un identificador del formato de archivo, que puede ser BMP, TIFF, JPEG, PNG, o AIPD.

Error Out: Contiene información del error en el VI.

File path: es la ruta de dirección completa, incluida la unidad, el directorio y el nombre de archivo a leer.

1.3 AQUISICION DE IMÁGENES CON NI-IMAQ PARA CAMARAS USB

1.3.1 NI IMAQ PARA CAMARAS USB

El módulo LabVIEW NI Vision es para todo individuo que desarrollan aplicaciones de visión artificial, en las siguientes secciones se describe los conceptos básicos e información para aplicaciones con “NI-IMAQ for USB Cameras”.

LabVIEW soporta una gran cantidad y variedad de cámaras ya sean estas webcams, microscópicas, escáneres y otras, todas están divididas en 2 grupos como lo son las cámaras analógicas y digitales, a continuación se muestra el menú del IMAQ USB en la figura 1.32.

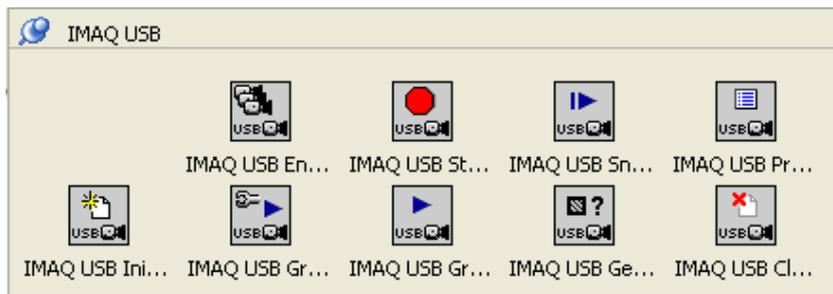


Figura 1.32 Menú IMAQ USB

FUENTE: LabVIEW (modificado el Autor)

NI-IMAQ para cámaras USB provee a través de su Asistente de Visión (Vision Assistant) 6 opciones distintas para adquirir imágenes:

- Adquisición de Imágenes desde una cámara seleccionada.
- Adquisición de Imágenes desde una cámara IEEE 1394.
- Adquisición de Imágenes desde una cámara IEEE 1394 o GigE seleccionada
- Adquisición de Imágenes desde una cámara USB seleccionada.
- Adquisición de RT: Adquiere imágenes de un dispositivo PXI que control LabVIEW en tiempo real.
- Simulador de Adquisiciones: usado para simular una adquisición real cargando imágenes de archivo.

Para esta Tesis, se optó por la adquisición con una cámara web conectada a través del puerto USB de la computadora cuyos mandos de adquisiciones disponibles en LabVIEW se detallan a continuación y en el orden necesario para realizar una adquisición de video, para una mejor comprensión véase la figura 1.34.

IMAQ USB Enumerate Cameras: sirve para enumerar las cámaras y dispositivos de video que se encuentran conectados al computador, este se lo utiliza con un VI de índice de arreglo para especificar el dispositivo.

IMAQ USB Init: configura la cámara USB y la prepara para la adquisición de la imagen

IMAQ USB Grab: Adquiere continuamente una imagen de la cámara seleccionada. Se utiliza este modo para adquirir imágenes a alta velocidad además que requiere usar siguientes dos VI:

- *IMAQ USB Grab Setup*: permite capturar la imagen en un buffer interno.
- *IMAQ USB Grab Acquire*: se lo utiliza para copiar la imagen almacenada en el buffer interno hacia el buffer en LabVIEW.

IMAQ USB Close : permite finalizar la adquisición luego que el programa termine de capturar las imágenes

IMAQ USB Snap: Permite adquirir una sola imagen de la cámara seleccionada (foto). Se utiliza este modo para adquirir un solo frame en el buffer y en capturas que no requieren mucha velocidad, el diagrama de bloques se indica en la figura 1.33

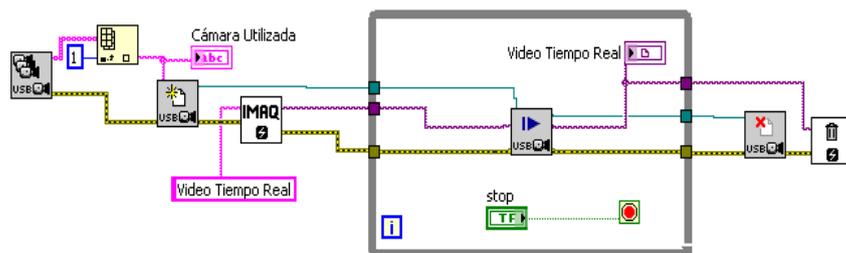


Figura 1.33 Adquisición de Imágenes Modo Snap

FUENTE: (El Autor)

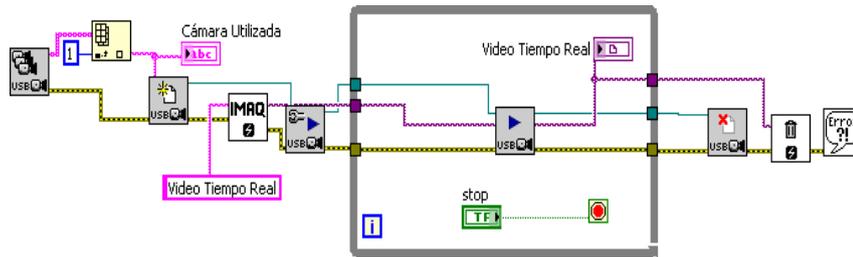


Figura 1.34 Adquisición de Imágenes Modo Grab
FUENTE: (El Autor)

1.4 PROCESAMIENTO Y ANALISIS DE IMÁGENES

1.4.1 PROCESAMIENTO DE IMÁGENES

El procesamiento de imágenes es la manipulación de una imagen inicial para obtener otra, para este proyecto se utilizó el instrumento virtual representado en la figura 1.35.

IMAQ Color BCGLookup: Aplica Brillo, contraste y corrección gama a varios planos de color separadamente.

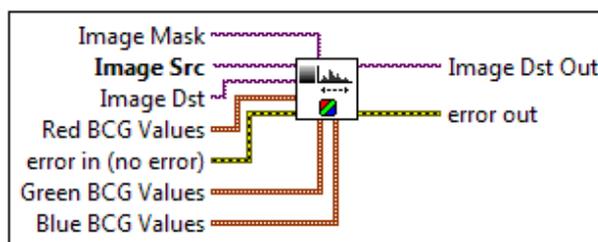


Figura 1.35 IMAQ Color BCGLookup

FUENTE: Ayuda de LabVIEW (modificado el Autor)

Dependiendo del tipo de filtro que apliquemos, obtendremos diferentes resultados de distinción de colores, la figura 1.36 visualiza los resultados de aplicar Color BCGLookup al video de un led encendido, se observa que detecta los objetos que presentan brillo y los presenta de color azul debido a la configuración aplicada.



Figura 1.36 Adquisición de video aplicado Color BCGLookup

FUENTE: (El Autor)

1.4.2 ANALISIS DE IMÁGENES

El análisis de imágenes permite la extracción de información de una imagen. Los Vis utilizados para este análisis, se detallan a continuación.

IMAQ Setup Learn Color Pattern: Define los parámetros a usar en la etapa de aprehensión de color. Este VI se debe utilizar antes del IMAQ Learn Color Pattern ya que introduce las configuraciones para la búsqueda, estableciendo el modo de aprendizaje, búsqueda y la saturación a ser considerada.

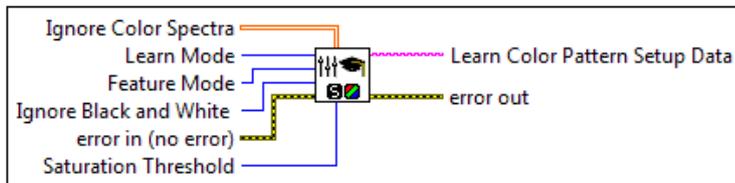


Figura 1.37 IMAQ Setup Learn Color Pattern

FUENTE: Ayuda de LabVIEW (modificado el Autor)

IMAQ Learn Color Pattern: crea una descripción de los parámetros de la plantilla que se va a buscar durante la etapa de emparejamiento de formas en el color, en este caso se ha elegido la imagen mostrada en la figura 1.38. Estos datos de descripción con añadidos a la imagen de la plantilla de entrada durante la búsqueda para que el color sea extraído de la plantilla y buscar en la imagen principal.



Figura 1.38 Imagen Patrón a ser analizada por *IMAQ Learn Color Pattern*

FUENTE: (El Autor)

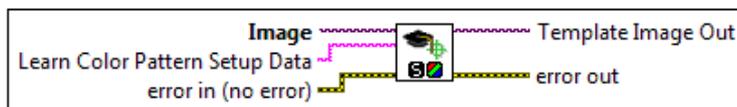


Figura 1.39 IMAQ Learn Color Pattern

FUENTE: Ayuda de LabVIEW (modificado el Autor)

IMAQ Match Color Pattern: Permite la búsqueda de un modelo en color, o la imagen de plantilla en color, en la entrada Image del VI determinando así los aciertos en la imagen fuente.

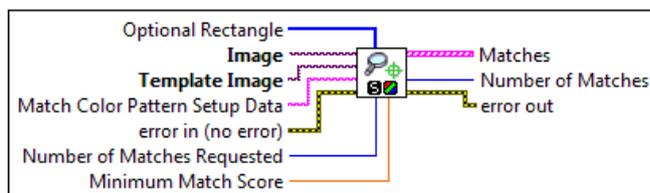


Figura 1.40 IMAQ Match Color Pattern

FUENTE: Ayuda de LabVIEW (modificado el Autor)

CAPÍTULO 2. INTERACCIÓN ELECTROMECAÁNICA ENTRE EL DEDAL Y EL SOFTWARE

2.1 DESCRIPCIÓN DE LA FUNCIONALIDAD DEL DEDAL

El dedal fue construido en dos etapas: La primera; conexión de leds, cableado, construcción de sócalos, conexiones para baterías y la segunda que es el sistema de ajuste del dedal para distinto tamaño de mano.

El lugar propicio para realizar e interactuar con el software puede ser una habitación que no contenga objetos brillantes o que reflejen la luz.

El sistema consta de 3 leds de alto brillo planos conectados en paralelo y en serie con una resistencia de 220 ohm para regular la corriente hacia los mismos. Estos están ubicados en línea recta con un espaciamiento entre ellos de 2.2cm. En términos anatómicos cada led está ubicado en la región media del dedo otro en la punta y un tercero en la parte posterior media del dedo, como se observa en la Figura 2.1.

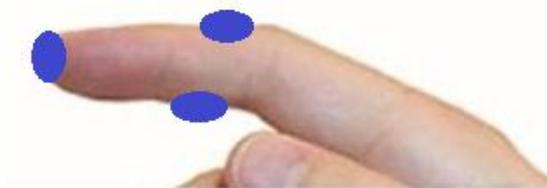


Figura 2.1 Ubicación de los leds de alto brillo en el dedal

FUENTE: (El Autor)

Estos componentes están conectados por medio de dos cables multipar a una batería de 3.7 voltios de tensión y 180mA de corriente.

Después de varias pruebas se decidió que el lugar adecuado en que la batería debe estar ubicada es en la muñeca del participante, debido a que esta zona es la que experimenta el menor movimiento y no causa molestias a la persona que utiliza el dispositivo.

Los cables que salen del dedal pasan a través de una banda cuya función es la de servir de soporte y nexo con una manilla regulable que permite asegurar el dispositivo evitando que el dedal resbale, brindando seguridad al estudiante y libertad para realizar cualquier clase de movimiento sin riesgo de estropearse, obsérvese la Figura 2.2

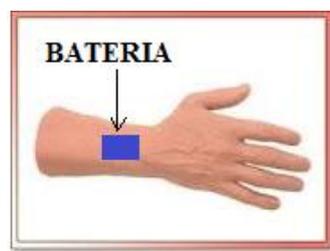


Figura 2.2 Ubicación de la batería

FUENTE: (El Autor)

Como se aprecia en la imagen anterior la batería está colocada a una distancia fija, debido a que los participantes de este proyecto pueden ser personas de distinta contextura y tamaño. Se optó por esta opción, puesto que para una mano de niño la batería estaría justo sobre la muñeca mientras que para una persona de mayor edad esta se ubicaría a nivel del dorso.

El sistema también consta de dos manillas regulables que dan firmeza y seguridad, estas deben ir unidas a la banda que atraviesa la mano y permite utilizar el dispositivo a cualquier persona, esto se aprecia en la Figura 2.3 en el literal a) se

observa como es la estructura de las manillas mientras que en el literal b) cómo deberían estar colocadas las manillas

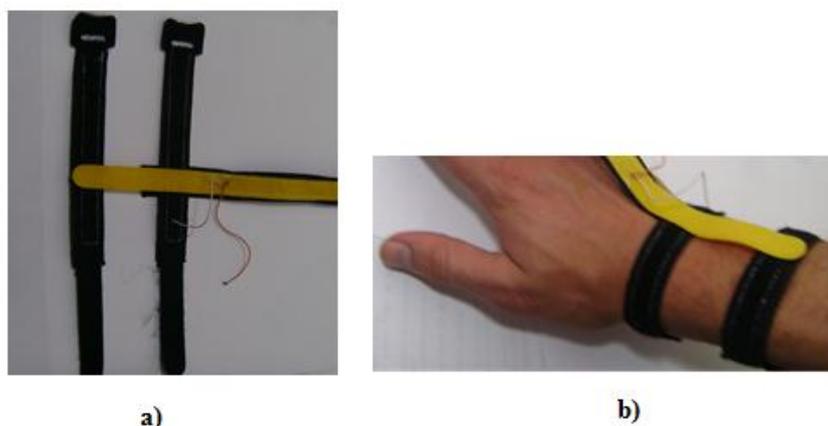


Figura 2.3 Mecanismo de ajuste de la manilla

FUENTE: (El Autor)

La presentación y la comodidad desempeña un papel fundamental cuando se desarrolla un producto, en base a ello se ha tomado en cuenta los siguientes puntos para la fabricación del dispositivo.

- Los cables están colocados entre el “adhesivo” y una cinta con la finalidad que no queden cables expuestos, puesto que este tiene una superficie bastante áspera todas los espacios que están en contacto con la piel son forrados a doble cara evitando así daños al participante tales como urticarias o alergias cuando se tiene piel sensible. Se muestra en la Figura 2.4.
- El tamaño de las manillas son; 20cm de largo x 1.5 de ancho c/u.
- El dedal tiene una longitud de 11 cm x 1.5 cm de ancho.
- La banda regulable que atraviesa la mano posee una longitud mínima de 5 cm y máxima de 17 cm x 1.5 cm de ancho.



Figura 2.4 Confección de forro a doble cara

FUENTE: (El Autor)

2.2 SELECCIÓN DEL AMBIENTE CONTROLADO

Un ambiente es controlado cuando cumple condiciones predeterminadas. Para este caso, se define como aquel que permite mejorar el desempeño del sistema de visión artificial, para controlar factores que podrían afectar al sistema, por ejemplo:

- Efectos de iluminación (sombras, deslumbramiento, reflejos etc.);
- Distancia entre la persona y el lente de la cámara.
- Objetos externos que aparezcan en la imagen y confundan al sistema.

El ambiente controlado permite desarrollar un sistema de visión artificial para extraer fácilmente las características de la imagen, haciéndole más rápido y exacto. Con este objetivo se busca un ambiente tal que el sistema solo se ocupe en analizar los objetos de interés (Luz), dejando de lado aspectos como el mejoramiento de la imagen, el cual es un proceso largo, requiere algoritmos complejos, y otros dispositivos que incrementarían considerablemente el costo.

En virtud de lo expuesto se ha elegido un ambiente controlado con las siguientes características:

- El lugar donde se coloque el sistema debe tener paredes de cualquier color, excepto blanco ni ningún tipo de pintura que refleje la luz, por ejemplo pintura látex.

- Puesto que se está trabajando con un proyector, para mejorar la visualización se requiere que la habitación no tenga demasiada iluminación ya que no permitirá apreciar de manera óptima las imágenes que se generan.
- De preferencia el lugar donde se encuentre ubicado el estudiante no debe contener objetos que reflejen la luz, tales como láminas, posters, relojes, espejos, etc.
- La vestimenta del participante no debe contener colores blancos.
- La iluminación artificial puede ser de cualquier tipo ya que esta no afecta el funcionamiento del dispositivo porque el software se encarga de seguir únicamente luces que estén directamente frente a la cámara y no le afecta la luminosidad del entorno.

Para una mejor comprensión en la Figura 2.5 se observa un ejemplo del entorno apropiado.

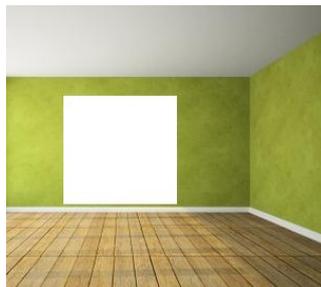


Figura 2.5 Entorno controlado para el funcionamiento del sistema

FUENTE: (El Autor)

Ventajas

- Hardware fácilmente transportable
- Ofrece libertad de movimiento al usuario
- Únicamente se ve afectada por las luces que están frente al lente de la cámara y no las del entorno.

Desventajas

- Detecta el color blanco y objetos que reflejan la luz, lo que confunde al sistema.

2.3 DESCRIPCIÓN DE LA FUNCIONALIDAD DE LA CÁMARA USB

La función de la cámara es capturar la imagen proyectada en el sensor para poder transferirla a un sistema electrónico.

El primer factor a considerar en un sistema de visión artificial es la cámara que se va a utilizar y para ello existen múltiples tipos de cámaras de acuerdo a la aplicación a desarrollar, entre ellas están; Cámaras con sensores CCD y CMOS, analógicas o digitales, cámaras lineales monocromáticas y de color, de espectros no visibles, Giga Ethernet, Firmware IEEE 1394, webcam, de alta resolución, de alta calidad, de alta velocidad, térmicas, infrarrojas, etc. Para este proyecto se investigaron únicamente las cámaras digitales y analógicas tanto de color como monocromáticas.[3]

La cámara a color tiene un costo más alto en relación a una monocromática además que requiere más algoritmos de procesamiento que una monocromática lo que afecta la respuesta en tiempo real. Figura 2.6.



Figura2.6 Comparación, imagen capturada con cámara monocromática y a color

FUENTE: (<http://princesadelalto.blogspot.com/2011/04/blanco-y-negro-o-color.html>)

2.3.1 CÁMARA DIGITAL Y ANALÓGICA

Un punto importante a tomar en cuenta cuando se elige una cámara es observar si el cable de conexión es apantallado para evitar ruido que dañe la calidad de la imagen. La principal ventaja de las cámaras digitales sobre las analógicas es que son más rápidas, porque transmiten los datos directamente a la tarjeta de adquisición. Otra ventaja es que las cámaras digitales pueden implementar bits de corrección de errores. Figura 2.7



Figura 2.7 Ejemplos de Cámara analógica y digital

FUENTE: (El Autor)

2.3.2 TIPO DE SALIDA DE VIDEO

Existen tres formatos que puede entregar una cámara de video: analógicos, digital monocromática y digital RGB.

Para este proyecto se optó por las cámaras digitales RGB puesto que poseen los colores fundamentales por separado lo que permite realizar el análisis de cada color.

2.3.2.1 FORMATO DE VIDEO

Para seleccionar esta característica de la cámara, se verifica el formato con el que trabaja la tarjeta de video. Por lo general existen dos formatos: el Europeo PAL a 50Hz y el americano NTCS 60Hz.

2.3.2.2 RESOLUCIÓN

Es un factor muy importante al momento de la compra ya que una cámara con mayor resolución se pueden distinguir detalles pequeños de la imagen, pero desafortunadamente esto implica mayor tiempo de procesamiento de la imagen e incremento del costo.

2.3.2.3 CARACTERÍSTICAS DEL SENSOR DE CÁMARAS DIGITALES MODERNAS.

La mayoría de cámaras digitales utilizan sensores CCD (Charge couple Device) y los CMOS (Complementary Metal Oxide Semiconductor).

Los dos sensores están formados por semiconductores de metal-óxido (MOS) y están distribuidos en forma de matriz. La función de estos sensores es de acumular una carga eléctrica en cada una de las celdas de la matriz (píxeles). Esta carga eléctrica depende de la cantidad de luz que incida sobre el mismo, cuanto más luz incida sobre el píxel mayor será la carga que éste adquiera, es decir, la carga que adquiere el píxel es directamente proporcional a la cantidad de luz.

Aunque las dos clases de sensores los CCD y CMOS funcionan de manera similar, existen diferencias entre las dos tecnologías.

2.3.2.3.1 SENSOR CCD

El sensor CCD convierte las cargas eléctricas de las celdas de la matriz en voltajes y entrega una señal analógica en la salida. En esta clase de sensor se hace una lectura de cada uno de los valores correspondiente a cada una de las celdas. Entonces, es esta información la que un convertidor analógico- digital traduce en forma de datos. La estructura interna del sensor es muy simple. El inconveniente de este sensor es la necesidad de un chip adicional que se encargue del tratamiento de la información proporcionada por el sensor, lo que provoca que el equipo sea grande y más costoso.

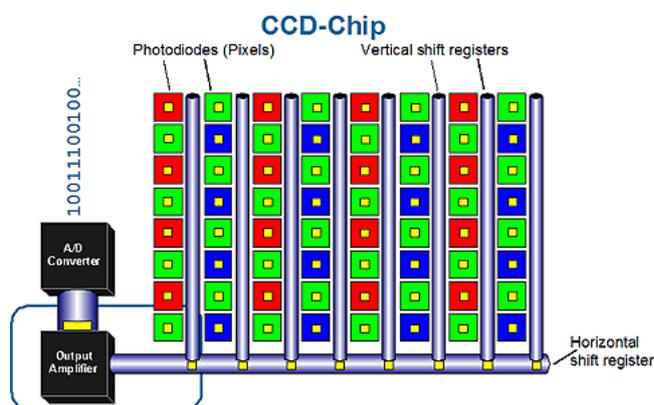


Figura 2.8 Estructura interna del sensor CCD

FUENTE: (<http://img.xatakafoto.com/2009/12/ccd.gif>)

El sensor CCD con respecto al sensor CMOS presenta las siguientes ventajas:

- Tolera mejor los extremos de la luz.
- Mejor control del ruido.

- Respuesta uniforme de los pixeles sometidos a un mismo nivel de excitación.

2.3.2.3.2 SENSOR CMOS

En este sensor cada celda (pixel) es independiente. La principal diferencia es que aquí la digitalización de los pixeles se realiza internamente en unos transistores que lleva cada celda, por lo que todo el trabajo se lleva a cabo dentro del sensor y no se hace necesario un chip externo que realice esta función, con esto se consiguen equipos más baratos y más pequeños.

La velocidad del CMOS es mayor a la del CCD debido a que todo el proceso se realiza dentro del propio sensor. Esta es una de las principales razones por las cuales estos sensores se utilizan en cámaras de alta velocidad de hasta 1000 fotogramas por segundo.

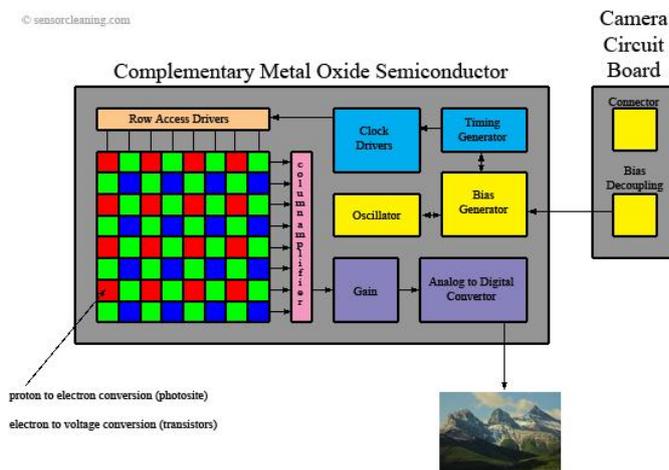


Figura 2.9 Estructura interna del sensor CCD

FUENTE: (<http://img.xatakafoto.com/2009/12/cmos.jpg>)

El sensor CMOS con respecto al sensor CCD presenta las siguientes ventajas:

- Son más sensibles a la luz, por lo que en condiciones pobres de iluminación se comportan mucho mejor.
- Control del fenómeno “blooming”. Este fenómeno se produce cuando un pixel se satura por la luz que incide sobre el y a continuación empieza a saturar a los que están alrededor

2.4 SELECCIÓN DE LA CÁMARA PARA LA APLICACIÓN

El software que utiliza LabVIEW 2009 posee una librería completa para leer cámaras USB, por ello escogió este tipo de conexión para nuestra cámara digital, además en vista que se busca que el sistema sea de bajo costo, se escogió una cámara de 30 fotogramas por segundo (fps) que son las más comunes en el mercado y su adquisición tiene un costo de aproximadamente 12 dólares.

Tomando en cuenta el ambiente controlado y tiempo de procesamiento se optó por la cámara web “D-Link DSB-C120 PC Camera”, como se observa en la Figura 1.50, la misma que posee las siguientes características:



Figura 2.10 D-Link DSB-C120 PC Camera

FUENTE: (<http://www.computech-ec.com/images/WebCam%20DLINK-C120.JPG>)

- Cámara Digital USB
- Trabaja en el plano RGB
- Tamaño de imagen 640x480 (requerido)
- 30Fotogramas por segundo
- Sensor CMOS

Dado que esta cámara posee auto contraste lo que afecta al programa de procesamiento de la imagen, esta función se la cancelara. Esto se lo explicara con mayor detalle en el capítulo siguiente.

2.5 MONTAJE DE LA CÁMARA

Puesto que el sistema es implementado en un institutito y por lo tanto no requiere la movilización de los equipos, la cámara está ubicada en un lugar fijo, sin embargo el momento que se quiera hacer transportable se requerirá de la adquisición de un trípode.

Para el primer caso la cámara se encuentra colocada sobre una estructura de aluminio que está sujeta a la pared por medio de tornillos con altura y giro regulable, con la finalidad de ajustarla a la altura del usuario y realizar el encuadre respectivo. En el segundo caso requerirá de una especie de trípode y de igual manera la cámara debe estar ubicada lo más cerca de la pared o estructura sobre la cual se esté proyectando, aprovechando el espacio disponible al máximo.



Figura 2.11 Montaje de la cámara fuera del aula.

FUENTE: (El Autor)

CAPÍTULO 3. DISEÑO E IMPLEMENTACIÓN

La idea de implementar un sistema de visión y el procesamiento de imagen para fines educativos, ya no es una idea futurista sin una solución del presente que facilita el desarrollo de las capacidades de aprendizaje y entendimiento de las personas, de manera especial en los niños cuyas mentes se encuentran en desarrollo y necesitan acostumbrarse al entorno actual en todos los aspectos. El proyecto que se presenta fue concebido a fin de dar solución a la necesidad de mejorar la calidad de la educación y de manera muy especial a un sector que prácticamente ha sido olvidado por la sociedad debido al desconocimiento y desinformación como son las personas con problemas de psicomotricidad, aprendizaje, entre otras. Sin embargo no se descarta la posibilidad que este software pueda ser implementado en las escuelas como un método más para la enseñanza.

3.1 DETERMINACIÓN DE PARÁMETROS DE DISEÑO

El desarrollo de un software versátil para el seguimiento de un patrón de luz es el principal objetivo a ser tomado en cuenta para el diseño del sistema. Las soluciones basadas en el procesamiento de imagen se emplean cada vez más, en especial en el momento de realizar la verificación y control de calidad de objetos, que se resuelve mediante técnicas clásicas de proceso de imagen y búsqueda de patrones de color, histogramas y algoritmos similares. El problema surge cuando bien por la naturaleza del ambiente o por la percepción visual, aparecen diferencias que a simple vista pueden considerarse leves y no afectan al objetivo principal en comparación a otras que si son consideradas como defectos graves.

Ante todo esta problemática a considerar, se ha desarrollado un sistema que permite la detección y seguimiento de una fuente luz teniendo en cuenta los siguientes parámetros:

- Tolerancia a reconocer luz en casi cualquier clase de entorno.
- Diferenciar la luz a seguir y los reflejos que se presenten de acuerdo con la plantilla pre-establecida, siguiendo el patrón si la luz coincide con la plantilla y omitiendo si no corresponde.
- Controlar la posición del puntero en cualquier dirección y velocidad de acuerdo al movimiento que siga el patrón (para este caso la luz).
- Debe ser capaz de detectar una fuente de luz pequeña que esté ubicada, máximo hasta una distancia de 6 metros y mínimo de 3 metros.
- Adaptarse automáticamente a cada elemento que constituye en sí el sistema: cámaras, computadora, led.

La aplicación se desarrolló bajo la plataforma NI, la cual es capaz de adaptarse para ser integrada dentro de un sistema de laboratorio así como un sistema de producción de pequeña escala. Su robustez, facilidad de uso y mantenimiento son elementos prácticamente indispensables como las funcionalidades que ofrece.

Los métodos de aprendizaje actuales de inspección visual para niños con problemas de psicomotricidad requieren de la presencia del tutor durante la clase o el ejercicio, surge la necesidad de un programa que de cierta manera independice al niño y se interese en realizar los ejercicios y actividades únicamente por la satisfacción que le representa utilizar el sistema, puesto que tiene un ambiente amigable y llamativo como se observa en la Figura 3.1

El sistema de visión trabaja de la siguiente manera:

1. Configuración de parámetros de comunicación entre la cámara y el CPU.
2. Calibración del sistema: inicialización del sistema. Selección y enfoque de la cámara a usar a fin de realizar el ajuste de la imagen de acuerdo al tamaño del niño.

3. Lectura de la plantilla pre-establecida a ser utilizada para la detección de la luz y establecer como valido o invalido.
4. Definición de los parámetros de búsqueda y tolerancia para ser considerados al momento de la búsqueda de coincidencias con cada patrón o imagen plantilla.
5. Configuración de parámetros de comunicación y acción de la fuente de luz (led) y el software.
6. Inicio del Sistema de detección. Una vez iniciado el proceso, el puntero empezara a moverse acorde a los parámetros establecidos anteriormente, el puntero seguirá de este modo tomando el control de la computadora deteniéndose únicamente cuando se presione la tecla ESC, seguido del botón de parada “STOP”.
7. Desarrollo de actividades: Automáticamente después de dar inicio el Sistema de detección se activara un menú para que el participante elija un juego he interactué con el software, estos juegos están ordenados de acuerdo al avance que tengan los participantes en su aprendizaje, las funcionalidades y su programación serán explicados más adelante, en este capítulo.



Figura 3.1 Pantalla principal del programa.

FUENTE: (El Autor)

3.2 DISEÑO DEL PATRÓN A RECONOCER

En la aplicación desarrollada, como puntos fundamentales se tomaron en cuenta los siguientes criterios:

- La imagen elegida como patrón debe contener dos colores. Un azul que será la luz y el negro que es el fondo y contorno de la plantilla.
- Al momento de crear la plantilla, el patrón a reconocer debe ser lo suficientemente grande para que LabVIEW la reconozca como imagen, caso contrario da errores debido a que no existe la suficiente información para procesar.
- La plantilla a desarrollar no debe tener falta de colores ni degradación del mismo, debe ser uniforme.

El diseño se enfocó en desarrollar una aplicación capaz de brindar al usuario la mayor precisión y control posible del puntero puesto que si la imagen era demasiado grande, tenía un área de búsqueda más amplia y lo desestabilizaba ya que buscaba continuamente la diferencia de color entre el azul y el negro, como se aparecía en la figura 3.2 y si está demasiado pequeño el programa no procesaba la imagen debido a la falta de información, enviando errores tales como “falta de contraste”, “falta de iluminación”.



Figura 3.2 Movimiento aleatorio del puntero a causa de una imagen plantilla grande

FUENTE: (El Autor)

3.2.1 CREACIÓN DE LA PLANTILLA

La creación de la plantilla se realizó en Paint y el tamaño apropiado se determinó en base a “prueba y error” hasta obtener un resultado óptimo.

La imagen fue creada en formato .bmp y guardada en la misma carpeta en donde se encuentra el programa principal, para una fácil localización.



Figura 3.3 Creación de la imagen plantilla en Paint

FUENTE: (El Autor)

3.2.2 SELECCIÓN DE PLANTILLAS

Para la selección de plantilla se hace uso del siguiente proceso.

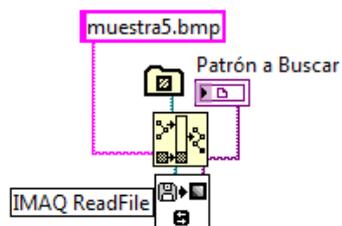


Figura 3.4 Proceso de selección de plantilla en el diagrama de bloques

FUENTE: (El Autor)

Una vez seleccionada la plantilla deseada, el visor “patrón a buscar” se muestra una vista previa de la misma, en el momento que se da inicio al programa, Figura 3.5



Figura 3.5 Imagen plantilla del patrón a reconocer en el panel frontal

FUENTE: (El Autor)

3.3 DISEÑO DEL PROGRAMA EN LabVIEW PARA EL SEGUIMIENTO DE LUZ.

El diagrama de bloques creado en LabVIEW permite tener el control del sistema en general. Una vez encendido el led del dedal, el programa comienza a detectar la posición de luz gracias a la cámara y la plantilla pre-seleccionada lo que permite que el puntero se mueva por toda la pantalla.

Esta pantalla cuenta con controles e indicadores a fin de facilitar al usuario el manejo de la herramienta. Se hace uso de 3 pestañas para evitar que el usuario deba desplazarse horizontal y verticalmente en el panel frontal de LabVIEW, el cual a medida que se presiona continuar en cada una de ellas se cambia automáticamente a la siguiente fase.

En la Figura 3.6 se observa que las etiquetas de las pestañas son:

- Presentación.
- Calibración y STOP
- Selección de menú.

Por motivos de estética se decidió ocultar las etiquetas de los paneles, y el contenido de cada una de estas se lo puede apreciar de mejor manera en el ANEXO A.



Figura 3.6 Pestañas implementadas en el Panel

FUENTE: (El Autor)

Algo de suma importancia a tomar en cuenta es: Cuando se ejecuta este programa se pierde el control del puntero por medio del mouse o pad puesto que el control de este lo realiza la luz frente a la cámara, por tal motivo se ha visto la necesidad de habilitar el teclado con todas sus funciones para comandar cualquier programa propio del sistema operativo.

3.3.1 PESTAÑA DE PRESENTACIÓN

La pestaña de presentación fue creada únicamente para permitir iniciar las variables, indicar la Universidad, el autor responsable del proyecto, un botón “CONTINUAR”, y darle así el aspecto de un juego de consola, la Figura 3.7 muestra lo explicado.

3.3.2 EL BOTÓN “CONTINUAR”

Se puede decir que este botón cumple la función principal de dar inicio a todo el programa. Para su explicación se lo ha dividido en 2 fases, Adquisición y Detección:



Figura 3.7 Aspecto de la pestaña de presentación
FUENTE: (El Autor)

3.3.2.1 ADQUISICIÓN

En esta etapa se configura los parámetros de la cámara con la que se está trabajando, estos son: tamaño de la imagen, brillo, contraste y colores. Es de suma importancia eliminar la función de auto contraste que posee la cámara, con esto se elimina el ruido que afecta a la etapa de detección, reduciéndose notablemente el tiempo de ejecución de la aplicación y mejorando la precisión del puntero.

Para este proyecto se está utilizando una cámara “D-Link DSB-C120 PC Camera” la misma que posee un módulo llamado “ArcSoft WebCam Companion” como se observa en la Figura 3.8, viene junto con el CD de instalación de la webcam y permite modificar los parámetros de la misma.



Figura 3.8 Programa ArcSoft WebCam Companion

FUENTE: (<http://4.bp.blogspot.com/Arcsoft+Webcam+Companion+3.JPG>)

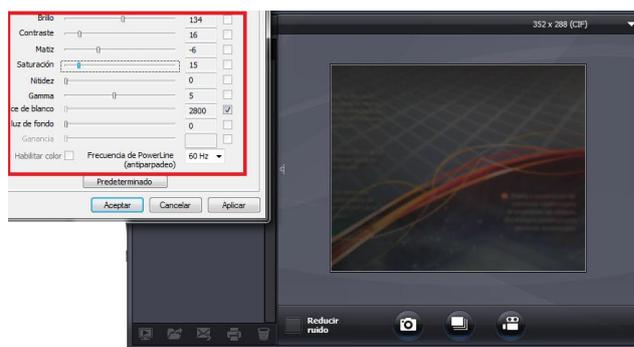


Figura 3.9 Valores de los parámetros a configurar

FUENTE: (El Autor)

En la Figura 3.10 se observa que el tamaño de la imagen que captura la webcam fue configurada para que sea de 352 x 288, la razón es debido a que las cámaras de bajo costo y tecnología ofrecen máximo esta resolución por lo que se trató de estandarizar para que el software funcione con cualquiera de estas.

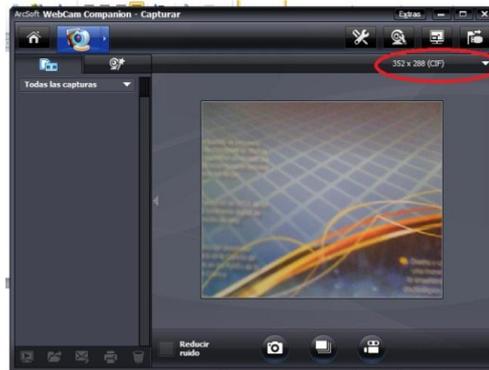


Figura 3.10 Configuración del tamaño de la imagen

FUENTE: (El Autor)

Ajustando a los valores de la Figura 3.9, la imagen que la cámara adquiere parece monocromática, pero en realidad está procesando una imagen RGB 32 con ganancias de colores bajos.

Para evitar el auto contraste y el ajuste automático de sombras es necesario calibrar los parámetros Contraste, Hue, Balance de blancos, etc., a los valores que se muestran en la Figura 3.9, mientras que el factor de brillo no se altera porque de este depende la adquisición de la imagen.

Se ha calibrado para que la cámara funcione a 60 Hz que es el estándar americano ampliamente utilizado en América latina, mientras que de 50Hz es para países Europeos como se explicó en líneas anteriores.

3.3.2.2 DETECCIÓN Y MOVIMIENTO

Con la finalidad que el sistema trabaje únicamente cuando esté presente un objeto (Fuente de luz) dentro del área de trabajo, se realiza un análisis matemático de los

pixeles dentro del ambiente controlado. Esta operación se ejecuta constantemente hasta que el usuario deje de usar la aplicación. Todo el proceso se efectúa en la imagen a color en el espacio RGB 32 bits. Puesto que el patrón a seguir se desplaza en coordenadas X e Y, y la resolución máxima que entrega la cámara es de 352 x 288, se vio la necesidad de multiplicar $352 * 2,9$ y $288 * 2,6$ con el objetivo de que el patrón se mueva por toda la pantalla la misma que tiene una resolución es $1280 * 1024$. Esa operación se observa en la Figura 3.11 en el diagrama de bloques de LabVIEW.

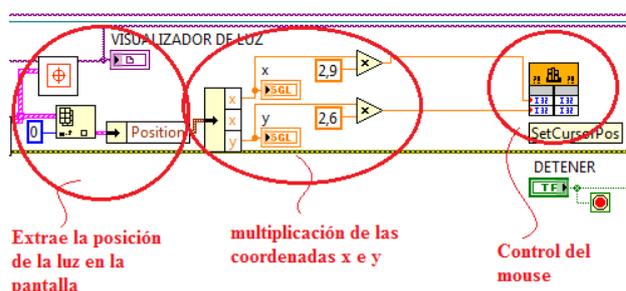


Figura 3.11 Multiplicación de las coordenadas X e Y en el panel de bloques y control del mouse

FUENTE: (El Autor)

Una vez obtenido este valor a la salida del multiplicador, para que el puntero se desplace por toda la pantalla es necesario utilizar el vi llamado “*Call Library Function Node*” que permite controlar todo periférico que se conecte a la computadora, esto lo hace gracias a que puede llamar cualquier librería del sistema operativo, en nuestro caso fue configurado para que controle únicamente la posición, cuya librería encargada de realizar esta función es la User32.dll, su configuración se muestra en la Figura 3.12

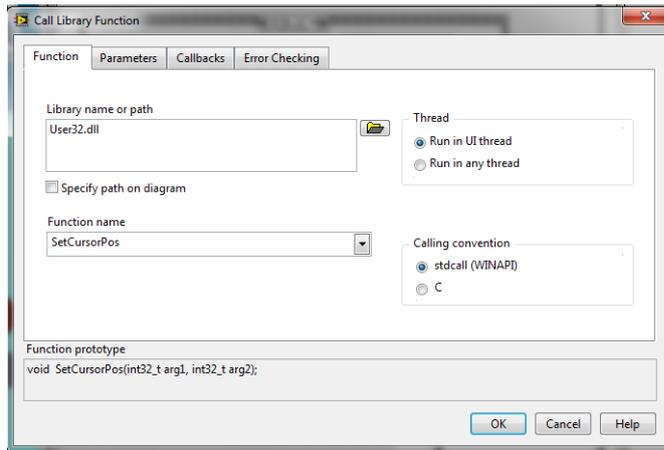


Figura 3.12 Configuración del vi “*Call library Funcion*” para control de posición del mouse

FUENTE: (El Autor)

3.3.3 PESTAÑA DE CALIBRACIÓN

Permite elegir la distancia apropiada que debe estar ubicado el estudiante respecto a la cámara para llegar a todos los puntos de la pantalla. En este panel se visualiza un mensaje “PARA CONTINUAR TOPAR LAS 4 IMÁGENES DE LAS ESQUINAS” que indica al estudiante lo que debe realizar para comenzar a jugar.

La solución más óptima para realizar esta calibración es ubicar 4 imágenes en las esquinas de la pantalla. El objetivo del estudiante es pasar el puntero sobre estas 4 imágenes para continuar con la siguiente etapa y pueda realizar los ejercicios didácticos programados. Observar Figura 3.13



Figura 3.13 Pestaña de Calibración y STOP

FUENTE: (El Autor)

3.3.4 PESTAÑA DE SELECCIÓN DE MENÚ

Esta pestaña contiene 12 imágenes que brindan una idea sobre los diferentes juegos que están disponibles, como se observa en la Figura 3.14, para ingresar a cualquiera de ellos es necesario colocarse sobre la imagen correspondiente durante unos 7 segundos puesto que no existe el control del clic derecho para este fin y para salir del mismo antes de que se termine se habilitó la tecla “ESC”. Se impuso este tiempo para que el participante tenga la libertad de elegir el juego que desee evitando así que ingrese por error a alguno de ellos.

La primera imagen es un juego desarrollado en LabVIEW con el objetivo de entrenar al participante y obtener una curva de aprendizaje de acuerdo a su evolución en el manejo del dispositivo, el desarrollo del mismo se explica en el punto 3.6.

Las 11 imágenes restantes permiten abrir el resto de juegos, estos fueron desarrollados en Flash Player dado que este software brinda mejores prestaciones cuando se trata del desarrollo animaciones e interactividad, en los siguientes párrafos se explicara con mayor detalle el uso de esta herramienta.

Con la finalidad de que este software pueda ser utilizado con más juegos que impliquen únicamente el movimiento del mouse a este panel se lo ha implementado un botón, el mismo que al estar posicionado sobre este 7 segundos desplegara una ventana en la cual podemos elegir cualquier juego descargado de internet o desarrollado por un tercero siempre y cuando tenga la extensión “.exe “.



Figura 3.14 Pestaña de Selección de menú

FUENTE: (El Autor)

3.4 DISEÑO Y PROGRAMACIÓN EN LABVIEW DE JUEGOS PARA NIÑOS

La programación de LabVIEW para realizar los juegos se desarrolló en base a consultas realizadas en la currícula académica para niños de básica y pre básica de acuerdo al Ministerio de Educación, profesores, psicólogos, y demás personal involucrado en el proceso de enseñanza a niños con problemas de psicomotricidad, obteniendo como resultado las siguientes pautas y consideraciones a fin de lograr los mejores resultados y cumplir con los objetivos del proyecto:

- Los juegos deben tener un ambiente amigable, llamativo y de fácil manejo.
- Antes de comenzar con los juegos de aprendizaje es necesario que los estudiantes pasen una fase de entrenamiento para que se acostumbren al manejo del cursor y comprendan su utilidad.
- Las aplicaciones deben ser desarrolladas y ubicadas en forma evolutiva, es decir comenzar con juegos básicos e ir incrementando la dificultad con el fin de desarrollar la mente del participante.
- No deben contener comentarios o acciones que reprochen o desmotiven al estudiante cuando se muestra un resultado, puesto que esta clase de niños son muy susceptibles.

3.4.1 DESARROLLO DEL JUEGO “ENTRENADOR”

A continuación se detallarán los pasos realizados para el desarrollo del juego Entrenador en LabVIEW, como no todos los juegos tiene las mismos requerimientos y sobre todo depende de la creatividad del desarrollador, lo que se explica en los siguientes párrafos puede servir de guía en el desarrollo de aplicaciones similares.

Para su desarrollo se utilizó un “*Tab*” con dos pestañas, en la primera está el nombre del juego y un botón “COMENZAR” como se muestra en la Figura 3.15, el mismo que al pasar el ratón sobre esta se habilita la segunda pestaña en donde se encuentra el juego. Su funcionamiento y programación se detalla a continuación.



Figura 3.15 Presentación de la pestaña de inicio del juego “ENTRENADOR”

FUENTE: (El Autor)

Este juego tiene como finalidad ayudar a que el participante mejore la precisión cuando maneja el dispositivo; Cuenta con un temporizador programado a 30 segundos y contador de errores para obtener una gráfica en Excel que sirve para indicar la evolución de la persona en el manejo del dedal, por tal motivo es necesario realizar esta prueba unas 10 veces para tener un buen indicador.

Como se observa en la Figura 3.16 el juego consiste en que el ratón llegue al queso en el menor tiempo posible a través del camino evitando en lo posible chocar con los bordes, en caso de que esto suceda el borde tocado se pintara de color rojo y se incrementara el contador de errores.

En caso de que se deseara salir y detener cualquier juego o regresar al menú se ha programado para que la tecla ESC cumpla con esta función gracias al uso del *“Event Estructure”* que como su nombre lo indica permite manejar eventos, en este caso se encarga del control del mouse y teclado. El diagrama de bloques en la Figura 3.17 explica lo antes mencionado.



Figura 3.16 Juego “ENTRENADOR”

FUENTE: (El Autor)

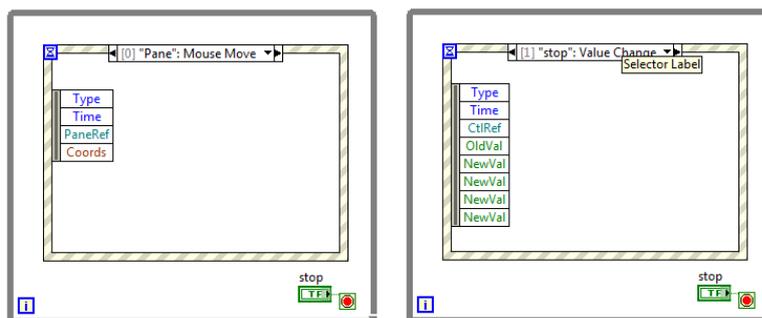


Figura 3.17 Configuración del *“Event Estructure”* para el control del mouse y teclado

FUENTE: (El Autor)

A continuación se detallaran los VI utilizados en LabVIEW para la realización de juegos con su respectiva explicación y funcionalidad.

Al utilizar un “*Event Estructure*” con la configuración mostrado en la figura 3.17 dentro de un “*While*”, permite obtener la posición del cursor en cualquier lugar de la pantalla ya sea dentro fuera de LabVIEW, al conocer este dato se puede manipular un booleano para que al momento en que el puntero se encuentre en determinada posición este se encienda, también se lo puede configurar para que cambie de pestaña, llame un subvi, incremente un contador, entre otras muchas funciones. En el ejemplo mostrado en la figura 3.18 el booleano cambiara a color rojo cuando el puntero se encuentre entre la posición 100-50 para el eje horizontal y entre 100 -200 para el eje vertical, es decir solo si el puntero está dentro de esta área el led se encenderá.

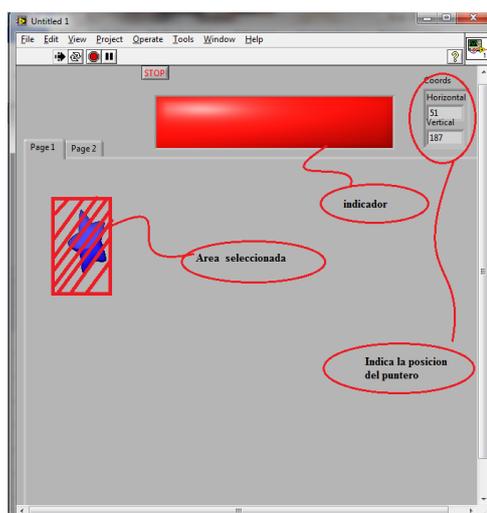


Figura 3.18 Panel frontal de un programa para reconocer posición del puntero

FUENTE: (El Autor)

Como se aprecia en la gráfica anterior existe un indicador con el nombre “*coords*” que muestra la posición del puntero en todo momento, por lo que podemos saber si

en realidad estamos dentro del rango especificado. En el diagrama de bloques de la Figura 3.19 se puede apreciar como esta implementado y una breve descripción de la función que realiza cada vi.

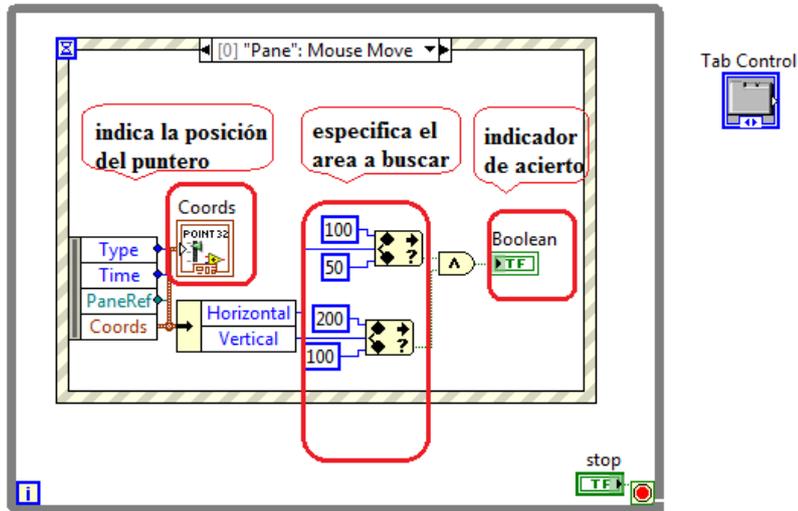


Figura 3.19 Diagrama de bloques de un programa para reconocer posición del puntero

FUENTE: (El Autor)

Para colocar imágenes en LabVIEW es necesario arrastrarlas desde cualquier ubicación hacia el área de trabajo para posteriormente recortarlas o modificar su tamaño, tomando en cuenta que no todos los formatos de imagen son aceptados por el programa sino únicamente aquellas que tiene las extensiones “*png*”, “*bmp*”, “*tif*” y “*gif*” cuya explicación se dio en el punto 1.2.4.2 de esta tesis.

En base a pruebas realizadas con diferente formato de imagen las que mejor se acoplan al programa son aquellas que tiene la extensión “*png*” cuya característica fundamental es que no existe pérdida de calidad y ofrece una muy buena compresión. Sus archivos suelen ocupar un poco más que el JPG pero con la

diferencia que permite transparencias, es decir no se visualiza el fondo de la imagen lo que le convierte en ideal para esta clase de aplicaciones, sin embargo, en caso de que se quisiera añadir imágenes animadas lo apropiado es utilizar las que tienen extensión “gif”.

3.5 DESARROLLO DE JUEGOS EN ADOBE FLASH PROFESSIONAL

En esta sección se verá de manera breve como desarrollar un juego en Adobe Flash Professional, específicamente el juego “CUENTA HASTA 5” cuya finalidad es enseñar a contar del 1 al 5.

Todas las imágenes para esta aplicación son vectoriales, (este y otros conceptos se detallan más adelante), las mismas que fueron desarrolladas y descargadas de librerías propias de Adobe.

El juego se desarrolló en 2 escenarios como se observa en la Figura 3.20, el primero que contiene la portada del juego y una flecha que dice empezar, el mismo que al pasar con el cursor sobre esta permite pasar al escenario 2 que es donde se encuentra el juego y todas las animaciones.



Figura 3.20 Escenarios del juego CUENTA HASTA CINCO

FUENTE: (El Autor)

Para una mejor comprensión a continuación se definirá algunos conceptos y procedimientos en el desarrollo de esta aplicación.

3.5.1 ADOBE FLASH PROFESSIONAL

Se trata de una aplicación de creación y manipulación de gráficos vectoriales con posibilidades de manejo de código mediante el lenguaje Action Script en forma de estudio de animación que trabaja sobre “fotogramas” y es utilizado para la producción y entrega de contenido interactivo. Es actualmente desarrollado y comercializado por Adobe Systems Incorporated y forma parte de la familia Adobe Creative Suite, su logo se muestra en la Figura 3.21.

Adobe Flash utiliza gráficos vectoriales, sonido, código de programa, flujo de vídeo y audio bidireccional. En sentido estricto, Flash es el entorno de desarrollo y Flash Player es el reproductor utilizado para visualizar los archivos generados con Flash.

Los gráficos vectoriales tienen la característica que requieren mucha menos memoria y espacio de almacenamiento que las imágenes de mapa de bits ya que se representan mediante fórmulas matemáticas en lugar de grandes conjuntos de datos.



Figura 3.21 Logo de Adobe Flash Player

FUENTE: (El Autor)

3.5.2 PROCEDIMIENTO PARA EL DESARROLLO DE JUEGOS

Para crear un juego flash, primeramente se crean gráficos vectoriales con herramientas de dibujo tales como el Adobe Illustrator y Adobe Photoshop de forma muy simple, posteriormente estos elementos se importan al documento de Flash. A continuación, se determina como y cuando se utilizarán cada uno de esos elementos para crear la aplicación que se tiene en mente.

Cuando se edita contenido en Flash, se trabaja en un archivo de documento de Flash. Estos documentos tienen la extensión de archivo “.fla” y está compuesto de cuatro partes principales:

El escenario es el espacio donde se muestran los gráficos, vídeos, botones y demás objetos durante la reproducción. Esta parte se observa en la Figura 3.22.

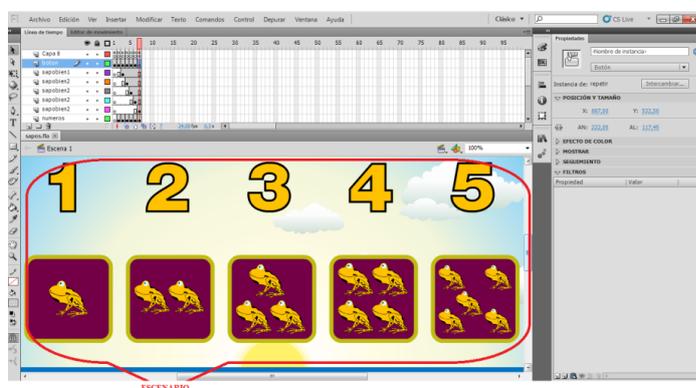


Figura 3.22 Delimitación del escenario de Adobe Flash Player

FUENTE: (El Autor)

La línea de tiempo es la parte del programa en donde el usuario indica a Flash cuándo desea que se muestren los gráficos y otros elementos del proyecto. También se utiliza para especificar el orden de capas de los gráficos en el escenario. Los gráficos de las capas superiores aparecen por encima de los gráficos de las capas inferiores. Obsérvese la Figura 3.23

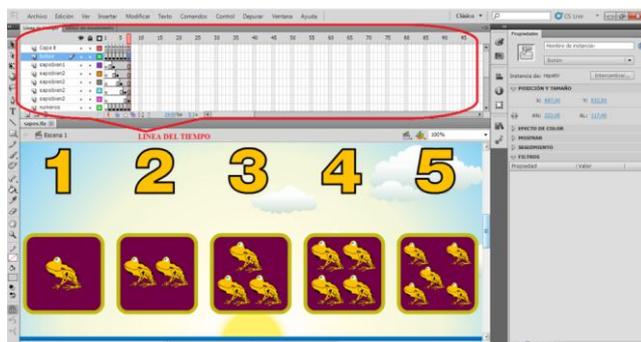


Figura 3.23 Delimitación de la línea de tiempo de Adobe Flash Player

FUENTE: (El Autor)

El panel Biblioteca es donde Flash muestra una lista de los elementos multimedia del documento de Flash.

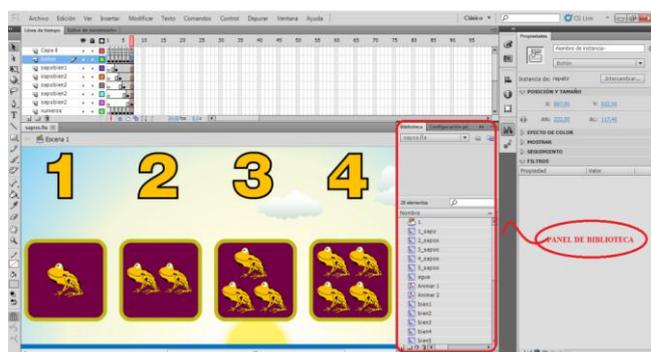


Figura 3.25 Delimitación del panel biblioteca de Adobe Flash Player

FUENTE: (El Autor)

Action Script es el código que permite añadir interactividad a los elementos multimedia del documento. Por ejemplo, se puede añadir código para que un botón muestre una nueva imagen o sonido cuando el usuario haga clic en el mismo. También se puede utilizar *Action Script* para añadir lógica a las aplicaciones o asignarle funcionalidad las teclas del cursor. Gracias a la lógica, la aplicación se comporta de distintas formas dependiendo de las acciones del usuario u otras condiciones. En la Figura 3.26 se muestra el código implementado para que cada vez que se pase con el mouse sobre la flecha repetir regrese al escenario 2 que son las condiciones iniciales.

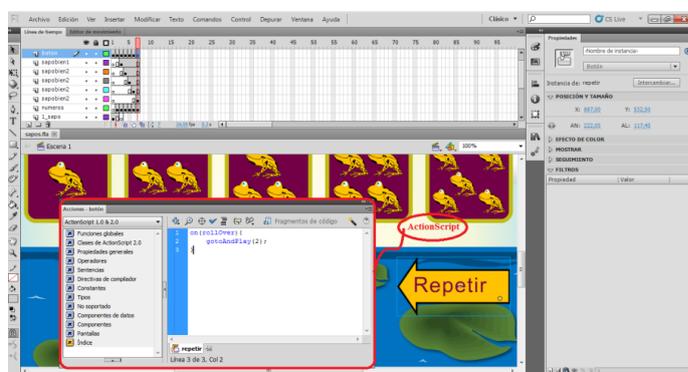


Figura 3.26 Delimitación del Action Script en Adobe Flash Player

FUENTE: (El Autor)

CAPITULO 4. PRUEBAS REALIZADAS Y PUBLICACION DE RESULTADOS

En este capítulo se muestran los resultados obtenidos al probar el algoritmo implementado en LabVIEW para el control del puntero, además de una gráfica que indica el tiempo aproximado que toma acostumbrarse al uso del dispositivo.

4.1 IDENTIFICAR EL COLOR LUZ APROPIADO

Una de las primeras pruebas realizadas fue escoger el tipo de iluminación que favorezca la captura de la imagen realizada por la cámara, se iniciaron las pruebas teniendo una idea que a mayor iluminación hay mas posibilidad de error puesto que los elementos que rodean a la habitación reflejan la luz y podría llevar a que el software funcione de forma incorrecta.

Después de probar en distintos escenarios con diferente cantidad de luz se determino que lo conveniente es evitar los lugares que contengan de fondo espejos o elementos refractivos, esto únicamente para darle mayor confiabilidad al software, puesto que el mismo puede funcionar en cualquier ambiente, sin importar la cantidad de luz que este presente.

En principio el Software funcionaba con cualquier color de luz que estaba presente, pero debido a que esto era un punto de fallo ya que identificaba cualquier luz que estaba frente a la cámara se opto por hacer que únicamente reconozca un color de luz determinado.

Para elegir el tipo de luz y color apropiado se tomo en cuenta que esta no debería ser común como por ejemplo la luz blanca o amarilla, sino un color rojo, azul o incluso

verde ya que estos colores generalmente no se ven en el ambiente. De esta manera se permitió configurar los ajustes necesarios para obtener buenos resultados.

4.1.1 DATOS OBTENIDOS EN BASE A PRUEBA Y ERROR

En base a prueba y error se probó con los tres colores determinando que el más apropiado es el color azul, este se encuentra en el rango de 450–495 nm del espectro electromagnético y es el color que la cámara con mayor facilidad capta, sin embargo también funciona con el color rojo que está en el rango de 620-750 nm, por lo que se podría concluir que únicamente capta los colores que están a los extremos del espectro y descarta los intermedios entre ellos el verde.

4.2 MANEJO DEL SISTEMA

En vista que se necesitaba un método para evaluar el desempeño del software se creó un juego llamado Entrenador cuyo desarrollo se explicó en el punto 3.4.1 de esta tesis. El mismo consiste en hacer que la imagen del ratón siga un camino y llegue hasta la meta que es la imagen de un queso, en el menor tiempo posible, evitando en lo posible chocar contra los muros del laberinto, obsérvese la Figura 3.16.

En este punto se contó con la colaboración de 3 participantes del I.P.C.A (Instituto de Parálisis Cerebral del Azuay) que no tenían ningún conocimiento previo sobre el funcionamiento del sistema, a los mismos que se les pidió que repitieran el juego unas 10 veces con el fin de obtener información y tabular los datos obtenidos en Excel, lo que posteriormente servirá para extraer gráficas de aprendizaje.

Los parámetros a tabular en Excel fueron: El numero de errores que cometía (entendido como error cada vez que el participante tocara el muro del laberinto) y el tiempo en segundos que le tomaba atravesarlo.

4.2.1 DATOS OBTENIDOS DEL MAJEJO DEL SISTEMA

Los resultados que se obtuvieron se detallan a la Tabla 4.1.

Número de intentos	Participante 1		Participante 2		Participante 3	
	Número de Errores	Tiempo empleado (seg)	Número de Errores	Tiempo empleado (seg)	Número de Errores	Tiempo empleado (seg)
1	26	28	26	25	24	23
2	28	26	24	23	24	22
3	19	22	22	21	23	20
4	17	18	21	20	19	15
5	15	20	18	18	16	14
6	14	18	15	15	15	13
7	12	15	12	12	15	13
8	10	13	10	10	14	12
9	9	10	10	9	11	10
10	7	9	8	9	10	8

Tabla 4.1 Valores obtenidos del desempeño de los 3 estudiantes del I.P.C.A

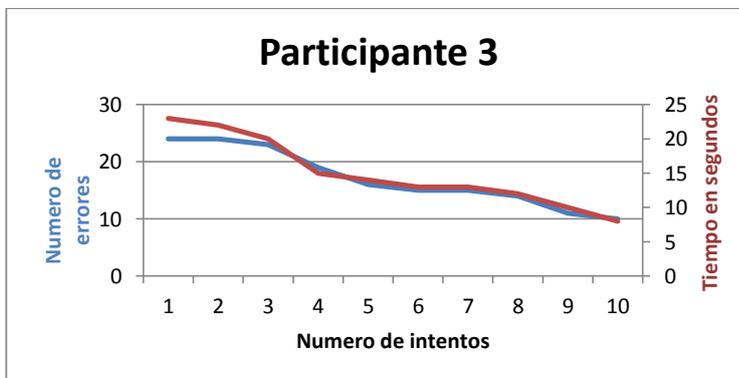
FUENTE: (El Autor)



a)



b)



c)

Figura 4.1 Graficas de aprendizaje de los 3 participantes del I.P.C.A en un tiempo máximo de 30 segundos

FUENTE: (El Autor)

4.3 ANÁLISIS DE RESULTADOS

Como se puede apreciar en la grafica 4.1 con referencia al primer participante se puede decir que en los dos primeros intentos el número de errores se incremento pero desde el tercer intento este disminuye y en el último punto prácticamente existe una mejora notable.

Con referencia al segundo y tercer participante se puede decir que tuvo el comportamiento esperado ya que a medida que se iba incrementando el número de intentos el número de errores disminuía.

En la Figura 4.1 se observan las graficas obtenidas por cada participante en relación al tiempo que le toma terminar el ejercicio, con un tiempo limite de 30 segundos. Se puede decir que a medida que se incrementan los intentos, el tiempo que le toma recorrer el laberinto disminuye, por lo que se podría decir que es cuestión de práctica obtener el máximo rendimiento del sistema

Si obtenemos una grafica promedio de los tres participantes como se observa en la Figura 4.2. En donde el número de errores esta expresado en porcentajes en el eje de las ordenadas, y el número de intentos en el eje de las abscisas se puede decir que esta curva es un indicador de la eficiencia del sistema.

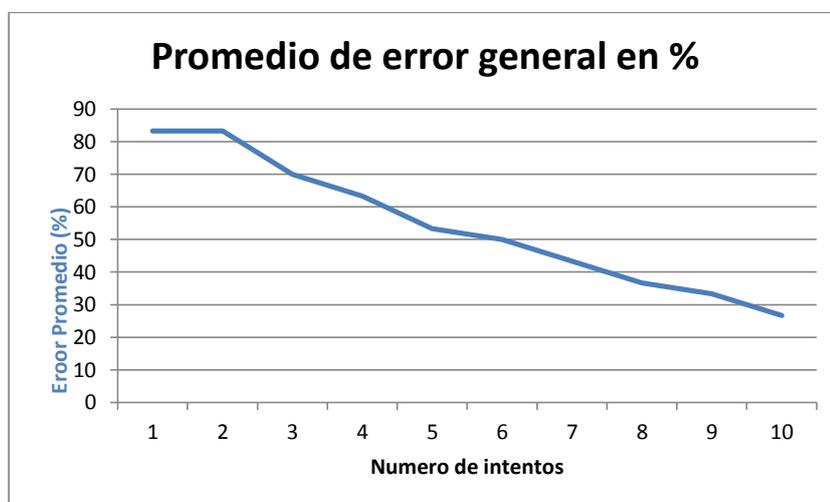


Figura 4.2 Grafica de aprendizaje promedio expresado en porcentajes

FUENTE: (El Autor)

CONCLUSIONES Y RECOMENDACIONES

Una vez desarrollado el hardware del proyecto, así como el software para el aprendizaje y desarrollo de la motricidad de niños con disfunción motora, y de los resultados de las pruebas realizadas es posible extraer las siguientes conclusiones y recomendaciones.

- ❖ Con la finalidad de obtener el óptimo funcionamiento del software es recomendable que el ambiente tenga poca iluminación, puesto que un sistema de iluminación correctamente seleccionado, reduce altamente el proceso de análisis de la imagen, la complejidad del sistema, y el tiempo de ejecución, de igual manera este tipo de iluminación colabora en gran medida cuando se trabaja con proyectores.
- ❖ LabVIEW es una herramienta de gran utilidad cuando se requiere desarrollo de software, control de periféricos tales como el mouse, teclado, joystick, permite realizar una programación ordenada y sencilla si comparamos con los lenguajes comunes como C++, java, etc. Además que posibilita añadirle una interfaz de fácil uso, amigable, con una presentación acorde a las necesidades que se requiera.
- ❖ El proceso de seguimiento del patrón depende del desempeño del algoritmo de procesamiento de la imagen implementado, dado que el pre-procesamiento resalta las características de la imagen que el clasificador necesita.
- ❖ El Toolkit de Visión de LabVIEW es una herramienta muy completa que contiene todas las operaciones necesarias para realizar cualquier tipo de procesamiento de imagen; esto permitió desarrollar totalmente la aplicación sin necesidad de un software adicional.

- ❖ Se investigaron los métodos para eliminar las partículas cercanas de una imagen; de los cuales se concluye que para aplicaciones donde existe demasiado ruido estos métodos son poco efectivos ya que al trabajar con video se necesita un procesamiento bastante rápido, por lo que se decidió modificar el control gama propio del software de la cámara para disminuir el brillo y reflejos de la habitación causante de la presencia de este ruido.

- ❖ Tomando como premisa el punto anterior, el funcionamiento de este software podría alcanzar una mejor calidad de procesamiento si se usara el módulo RIO de National Instruments, el cual permite trabajar el procesamiento de imágenes en tiempo real debido a que puede trabajar con cámaras rápidas como las SMART, en donde el procesamiento de imágenes prácticamente es en tiempo real, sin embargo la adquisición de este módulo y cámara es costoso si se compara con los elementos utilizados para este proyecto, ya que una de sus características principales es su bajo costo.

- ❖ En cuanto al manejo del software se puede concluir que cualquier persona que sea capaz de mover una de sus manos puede controlar el dispositivo perfectamente después de que haya adquirido suficiente práctica, como lo demuestran las graficas de aprendizaje obtenidas e incluso obtener otros parámetros que pueden permitir visualizar la evolución del estudiante en cuanto a su actividad Psicomotriz.

- ❖ Puesto que este proyecto va dirigido a una parte vulnerable de nuestra sociedad, cuando se realizaron pruebas, hay que tener especial cuidado cuando se trabajan con personas con discapacidad Psicomotriz, y tomar las debidas consideraciones, tales como:
 - Explicarle con detenimiento y con palabras adecuadas como manejar el dispositivo para no herir susceptibilidades.

- El dispositivo que se le coloca en la mano no debe causarle molestias.
- La adquisición de este proyecto debe ser económica, pues la mayoría de personas que se educan en este instituto son de bajos recursos.

- ❖ Es imprescindible la calibración previa de la cámara cuando se trabaja con sistemas de visión artificial, por ejemplo hay que deshabilitar las funciones automáticas de la cámara como el auto compensación de luz, auto contraste, mirror, todo esto con el fin de mejorar el tiempo de procesamiento de la imagen en LabVIEW.

- ❖ En un trabajo futuro se recomienda que este software tenga la funcionalidad de trabajar con dos personas simultáneamente, puesto que muchas veces el tutor desea interactuar con el estudiante o simplemente que trabajen dos estudiantes a la vez.

- ❖ El desarrollo de juegos puede ser realizado en Adobe Flash Player por profesionales que sepan las necesidades de los estudiantes, pues esta es una herramienta que permite controlar el movimiento del puntero por medio de luz tomando en consideración que no existe el control de los clics desde el dedal.

REFERENCIAS BIBLIOGRÁFICAS

- [1] TELLO PALADINES Denisse Elena, *Aplicación de visión con LabVIEW para la detección de frascos de distinto color*, Tesis Escuela Superior Politécnica del Litoral. Guayaquil – Ecuador, 2009.
- [2] BENÍTEZ R. Julio César, *Implementación de la estación de visión artificial y control de calidad del laboratorio CIM 2000: vi 2005*, Tesis Escuela Politécnica Del Ejército, Sangolquí – Ecuador, 2010.
- [3] CHIGUANO RODRIGUEZ Edwin Fabián , *Diseño e implementación de un sistema traductor de lenguaje de señas de manos a un lenguaje de texto mediante Visión Artificial en un ambiente controlado* , Tesis Escuela Politécnica Nacional, Quito- Ecuador, 2011.
- [4] ORDOÑEZ ALQUINGA Edison Paul, *Diseño e implementación de un sistema de clasificación de rosas aplicando Visión Artificial con LabVIEW*, Tesis Escuela Politécnica Nacional, Quito- Ecuador, 2011.
- [5] HIDALGO JACOME Victoria Alexandra, *Implementación de un sistema de Autenticación Biométrica Basado en huellas digitales*, Escuela Superior Politécnica de Chimborazo, Riobamba –Ecuador, 2010.
- [6] TRAVIS J., KRING J., *LabVIEW for everyone*, 1^{ra} Edición, Prentice Hall, Julio 2007.
- [7] GONZALES. R.; WOODS R. *Tratamiento digital de imágenes.*, 2^{ra} Edición, Addison-Wesley/Da de Santos, 2012
- [8] National Instruments, *LabVIEW-User Manual*, Edition Part Number 320999E-01, April 2003
- [9] Juan ALBORNOZ CARRASCO, *Guía de Iniciación en LabVIEW 7.1*, Universidad de la Frontera Facultad de Ingeniería, Ciencias y Administración Departamento de Ingeniería Eléctrica, 2008.

[10] *NI-MAQ for USB cameras*, National Instruments, Texas, 2009.

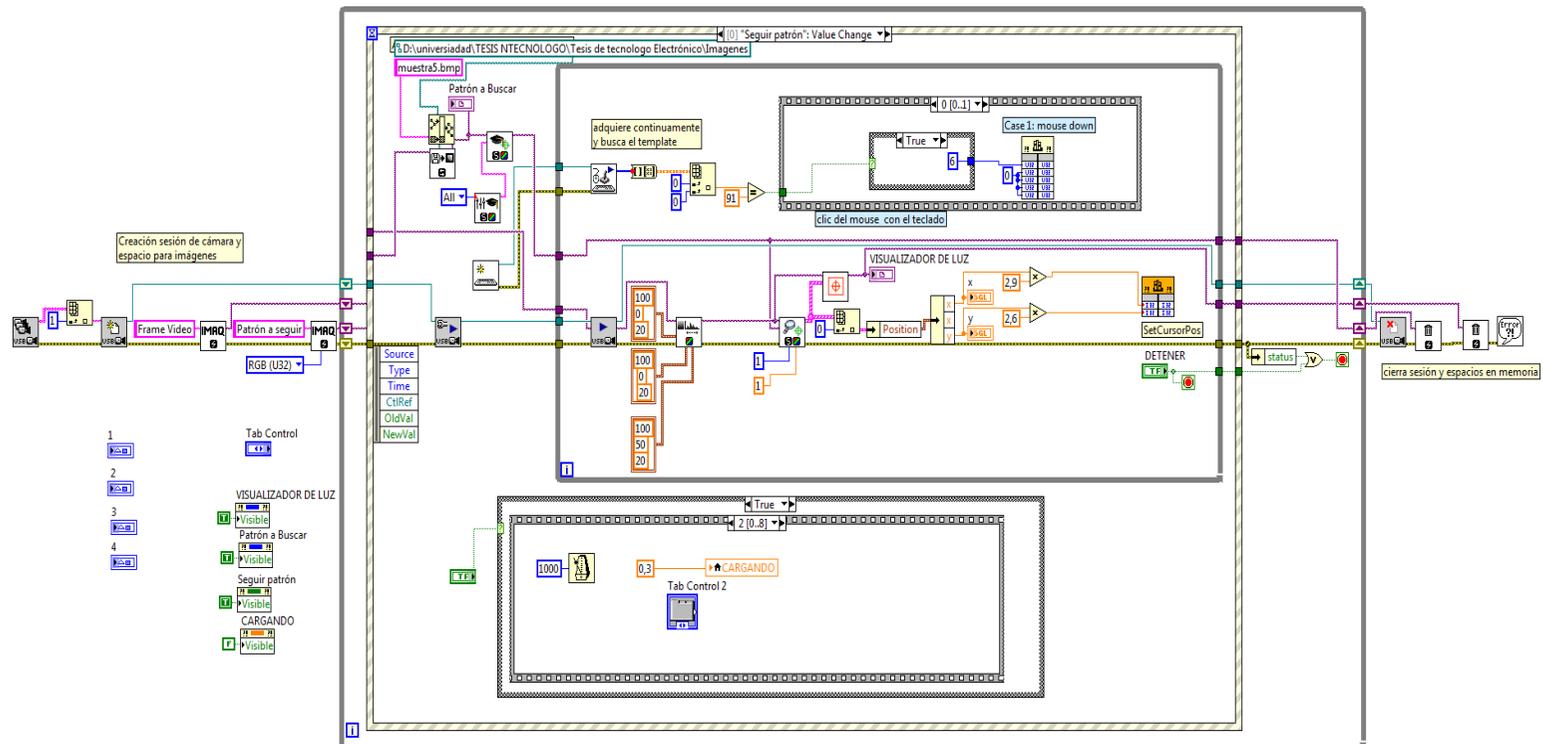
[11] Consejo Nacional de Discapacidades, Información en línea disponible en www.conadis.gob.ec/estadisticas.htm#estadis. Mayo, 2012.

[12] Guía de Iniciación en LabVIEW-Herramientas, Información en línea disponible en www.ni.com/gettingstarted/labviewbasics/esa/tools.html

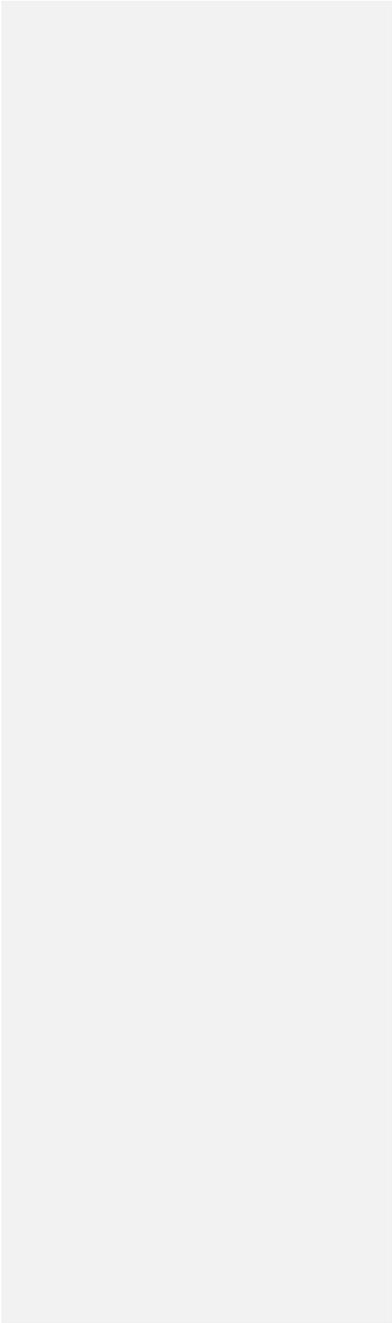
[13] Imágenes de Mapa de Bits, información en línea, disponible en www.revista.unam.mx/vol.6/num5/art50/art50-2.html

ANEXO 1

Diagrama de bloques implementado en LabVIEW para el reconocimiento de color, seguimiento de patrón y habilitación del teclado.



Panel Frontal pestaña de presentación



Panel Frontal pestaña de calibración y stop

PRESENTACION CALIBRACION Y STOP SELECCION DE MENU

PARA CONTINUAR TOCAR LAS 4 IMAGENES DE LAS ESQUINAS EN EL ORDEN INDICADO

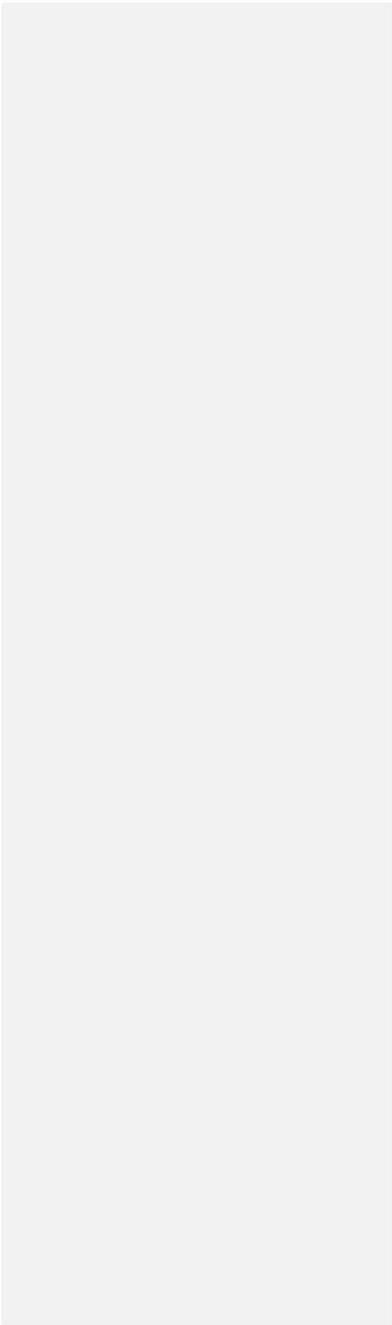


1 2

3 4

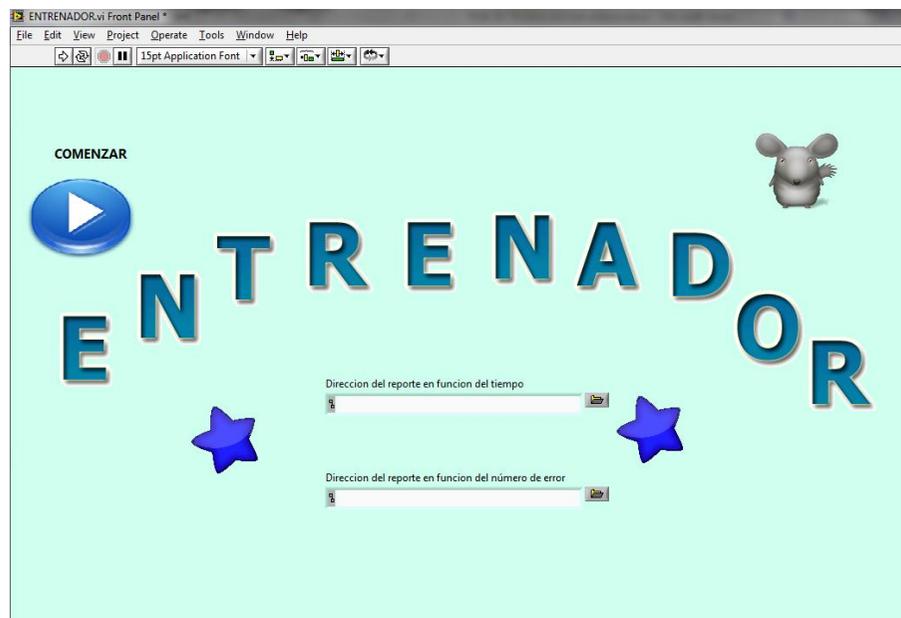


|

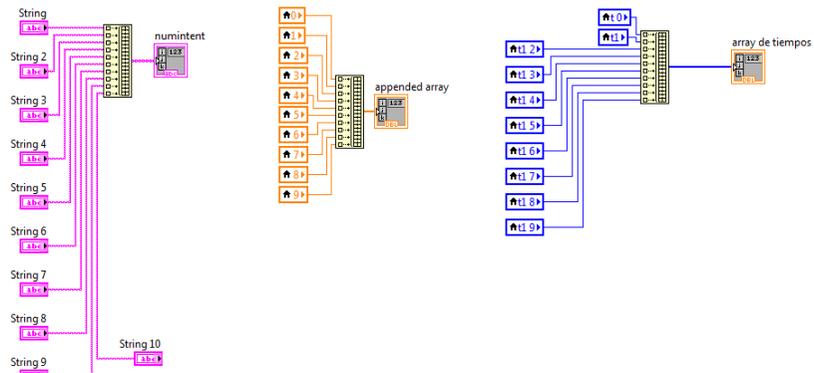
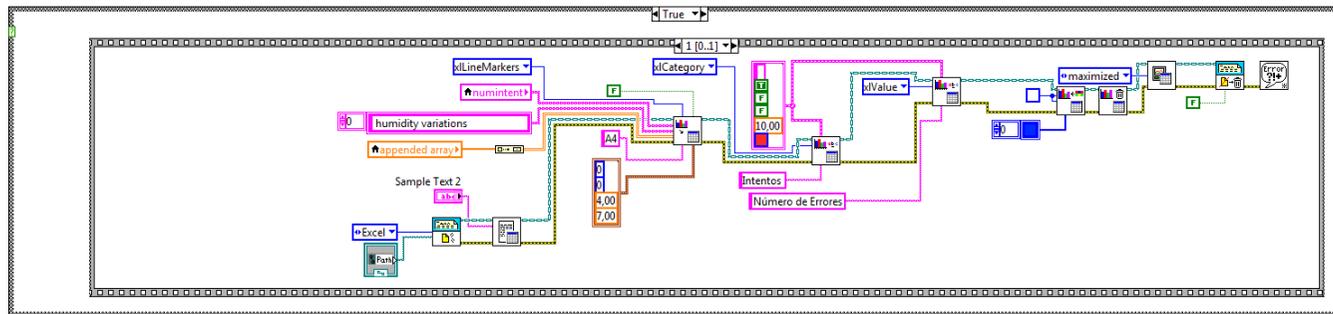


ANEXO 2

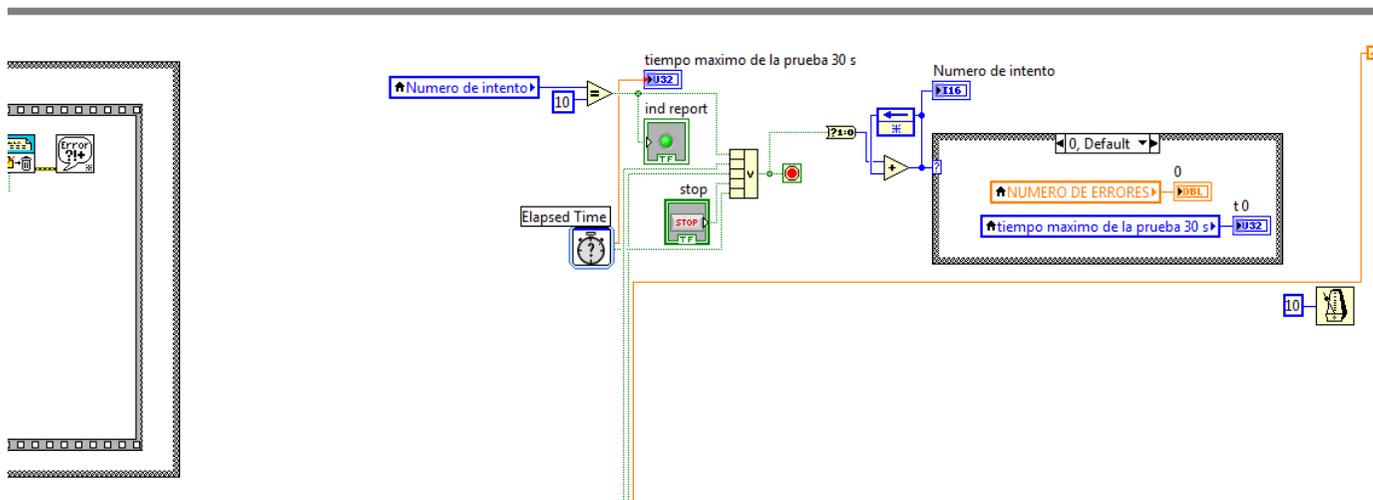
Presentación del Juego entrenador.



Bloque encargado de escribir datos Excel y obtener gráficas



Bloque que activa y desactiva la ventana junto con el contador de errores



Bloque encargado de reconocer la posición de la imagen y activar los boléanos

