



**INSTITUTO POLITÉCNICO NACIONAL**

**ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y  
ELÉCTRICA**



**UNIDAD PROFESIONAL “ADOLFO LÓPEZ MATEOS”  
ZACATENCO**

**“SISTEMA DE SEGURIDAD BIOMÉTRICO FACIAL  
UTILIZANDO ANÁLISIS WAVELET DISCRETO”**

**REPORTE TÉCNICO**

**PARA OBTENER EL TÍTULO DE:  
INGENIERO EN COMUNICACIONES Y ELECTRÓNICA**

**PRESENTAN:**

**RAMIREZ PATRICIO ABIGAIL JUANITA  
SAAVEDRA CHAVEZ MANUEL ROBERTO**

**ASESORES:**

**DR. MARIO JIMENEZ HERNÁNDEZ  
ING. GERARDO SUÁREZ**

**CIUDAD DE MÉXICO, MAYO 2021**

**INSTITUTO POLITÉCNICO NACIONAL**  
**ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELÉCTRICA**  
**UNIDAD PROFESIONAL “ADOLFO LÓPEZ MATEOS”**

**REPORTE TÉCNICO**

QUE PARA OBTENER EL TÍTULO DE INGENIERO EN COMUNICACIONES Y ELECTRÓNICA  
POR LA OPCIÓN DE TITULACIÓN CURRICULAR  
DEBERA (N) DESARROLLAR C. ABIGAIL JUANITA RAMIREZ PATRICIO  
C. MANUEL ROBERTO SAAVEDRA CHAVEZ

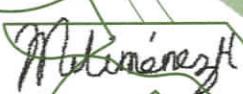
**“SISTEMA DE SEGURIDAD BIOMÉTRICO FACIAL UTILIZANDO ANÁLISIS WAVELET DISCRETO”**

EL OBJETIVO DE ESTE TRABAJO ES EL DE BRINDAR UNA HERRAMIENTA ÚTIL Y EFICAZ PARA UN CONTROL ESTADÍSTICO DE LA POBLACIÓN BASADO EN UN RECONOCIMIENTO FACIAL CONTANDO CON UNA BASE DE DATOS PARA EL ALMACENAJE Y COMPARACIÓN DE DATOS DE VITAL IMPORTANCIA PARA FUTURAS REFERENCIAS O INVESTIGACIONES EN EL ÁMBITO DE LA INVESTIGACIÓN CRIMINALÍSTICA.

- ❖ MARCO TEÓRICO
- ❖ DESARROLLO
- ❖ PRUEBAS Y RESULTADOS
- ❖ CONCLUSIONES

CIUDAD DE MÉXICO, A 21 DE MAYO DE 2021.

ASESORES

  
DR. MARIO JIMÉNEZ HERNÁNDEZ

  
ING. GERARDO SUÁREZ

  
M. EN C. ITZALÁ RABADÁN MALDA  
JEFA DE LA CARRERA DE INGENIERÍA  
EN COMUNICACIONES Y ELECTRÓNICA



## Autorización de uso de obra

**Instituto Politécnico Nacional**

**Presente**

Bajo protesta de decir verdad los que suscriben **ABIGAIL JUANITA RAMIREZ PATRICIO Y MANUEL ROBERTO SAAVEDRA CHAVEZ**, manifestamos ser autores y titulares de los derechos morales y patrimoniales de la obra titulada "**SISTEMA DE SEGURIDAD BIOMÉTRICO FACIAL UTILIZANDO ANÁLISIS WAVELET DISCRETO**", en adelante "**La Tesis**" y de la cual se adjunta copia *en dos cd's*, por lo que por medio del presente y con fundamento en el artículo 27 fracción II, inciso b) de la Ley Federal del Derecho de Autor, otorgamos al **Instituto Politécnico Nacional**, en adelante **EI IPN**, autorización no exclusiva para comunicar y exhibir públicamente total o parcialmente en medios digitales o en cualquier otro medio; *para apoyar futuros trabajos relacionados con el tema* de "**La Tesis**" por un periodo de **1 año** contado a partir de la fecha de la presente autorización, dicho periodo se renovará automáticamente en caso de no dar aviso expreso a **EI IPN** de su terminación.

En virtud de lo anterior, **EI IPN** deberá reconocer en todo momento nuestra calidad de autores de "**La Tesis**".

Adicionalmente, y en nuestra calidad de autores y titulares de los derechos morales y patrimoniales de "**La Tesis**", manifestamos que la misma es original y que la presente autorización no contraviene ninguna otorgada por los suscritos respecto de "**La Tesis**", por lo que deslindamos de toda responsabilidad a **EI IPN** en caso de que el contenido de "**La Tesis**" o la autorización concedida afecte o viole derechos autorales, industriales, secretos industriales, convenios o contratos de confidencialidad o en general cualquier derecho de propiedad intelectual de terceros y asumimos las consecuencias legales y económicas de cualquier demanda o reclamación que puedan derivarse del caso.

Ciudad de México., a 15 de Junio de 2023.

**Atentamente**

**ABIGAIL JUANITA RAMIREZ PATRICIO**

**MANUEL ROBERTO SAAVEDRA CHAVEZ**

## Índice general

1. Introducción	7
1.1 Breve historia de la investigación criminal	7
1.2 Planteamiento del Problema	8
1.3 Propuesta de solución	9
1.4 Aportaciones	9
1.5 Justificación	9
1.6 Hipótesis	10
1.7 Objetivo General	10
1.8 Objetivos Específicos	10
1.9 Alcances del Trabajo	10
2. Estado del Arte	11
2.1 Imagen digital	11
2.2 Formato de Imágenes	11
2.2.1 Windows BMP	11
2.2.2 Archivo de cabecera	12
2.2.3 Imagen de cabecera	12
2.2.4 Paleta de Colores	13
2.2.5 Información de Pixel	13
2.3 Compresión	14
2.3.1 Formato	15
2.3.2 Formato JPEG	15
2.3.3 Modos de compresión	16
2.3.4 Progresivo	16
2.3.5 Jerárquica	16
2.3.6 Sin pérdidas	17
2.4 Sistemas de colores	17
2.4.1 RGB	17
2.4.2 Escala de grises	19
2.5 Comparativa entre sistemas biométricos	19
2.5.1. Métodos de detección de rostros	20
2.5.2 Métodos de detección facial	21
2.6 Detector de caras: Algoritmo de Viola Jones	21
2.6.1 Integral de la imagen	21
2.7 Características de Haar	21
2.8 Transformada de Wavelet discreta	22
2.9 Transformada de Haar Wavelet	23
2.9.1 Transformada Haar de dos dimensiones como un producto de matrices	24
2.10 Características útiles para la compresión de imágenes	25
2.11 Histograma del Gradiente Orientado	27
2.12 Máquina de vectores de soporte (SVM)	28
2.13 Códigos de salida de corrección de errores (ECOC)	31
3. Propuesta de solución	33
3.1 Desarrollo del programa	34
3.1.1 Carga de base de datos	37

3.1.2 Etapa de captura	39
3.1.3. Etapa de compresión	40
3.1.4 Detección de rostros	41
3.2 Desarrollo Interfaz gráfica	42
4. Pruebas y Resultados	58
4.1 Prueba con entorno controlado	58
4.1.1 Prueba con entorno controlado sujeto 2	60
4.2 Prueba con entorno no controlado	61
5. Conclusiones	63
5.1. Aportaciones	63
5.2. Trabajos futuros	63
5.3 Conclusiones	64
6. Referencias Bibliográficas	65

## Índice de figuras

2.1 Estructura archivo BMP. [CIF00]	11
2.2 Formato para 16 bits RGB. [POL1]	14
2.3 Modos JPEG .[CIF06]	16
2.4 Modelo de color RGB.[CIF07]	17
2.5 Representación en escala de grises.[GRA1]	19
2.6 Ejemplo de características de rectángulo mostradas en relación con la ventana de dirección adjunta.[MAI1]	23
2.7 Ejemplo de imagen en blanco y negro para su análisis	29
2.8 Análisis de píxeles	29
2.9 Orientación de flechas	30
2.10 Análisis de brillo de las imágenes	30
2.11 Detección de rostros	31
3.1 Funcionamiento general del prototipo	33
3.2 Diagrama de flujo	35
3.3 Contenido del objeto detector	37
3.4 Comparación de almacenamiento entre imagen comprimida(a) imagen sin compresión(b).	41
3.5 Detección por Viola Jones	42
3.6 Menú Principal Interfaz Grafica	43
3.7 Menú Registro	43
3.8 Botón Captura	45
3.9 Botón captura con detección de rostros	46
3.10 Guardado de Fotografía en Base de Datos	46
3.11 Registro exitoso de usuario	47
3.12 Interfaz Sistema de Seguridad	48
3.13 Sistema de seguridad con un usuario	49
3.14 Acceso del usuario con seguridad	49
3.15 Intento de Ingreso con falla de seguridad	50
3.16 Interfaz Búsqueda de usuarios	51
3.17 Selección de Imagen	51
3.18 Interfaz secundaria (ingresar Foto)	52
3.19 Identificación del sujeto mediante imagen	52
3.20 Interfaz tomar captura	53
3.21 Fotografía tomada con éxito para la búsqueda	54
3.22 Botón Continuar	54
3.23 Identificación de usuario mediante captura de fotografía en tiempo real	55
3.24 Interfaz base de usuarios	56
3.25 1er Usuario registrado en base de datos	56
4.1 Imágenes capturadas del usuario 1 entorno controlado	58
4.2 Imágenes captadas del usuario 2 en un entorno controlado	60
4.3 Imágenes captadas con entorno no controlado	62

## **Índice de tablas**

2.1 Estructura BITMAPFILEHEADER.[CIF01]	12
2.2 Estructura BITMAPINFOHEADER.[CIF02]	12
2.3 Estructura BITMAPCOREHEADER.[CIF03]	13
2.4 Estructura RGB Doble.[CIF04]	13
2.5 Estructura RGB Triple.[CIF05]	13
2.6 Valores validos RGB.[DIP2]	18
2.7: Diferentes características de los sistemas biométricos	20
4.1: Resultados de usuario 1 con entorno controlado	59
4.2: Resultados de usuario 2 con entorno controlado	61
4.3: Resultados con entorno no controlado	63

## **Índice de gráficas**

4.1 : Resultados prueba 1	59
4.2 : Resultados prueba con sujeto 2	61
4.3 Resultados de prueba entorno no controlado	63

# Capítulo 1

## Introducción

*El reconocimiento facial es una solución biométrica que emplea un algoritmo automático para verificar y reconocer la identidad de una persona en función de sus características fisiológicas. El presente capítulo describe primero los antecedentes históricos en la investigación criminal, para así introducir el reconocimiento facial como herramienta de identificación.*

*Posteriormente se describe el planteamiento del problema del reconocimiento facial y la base de datos. Para así lograr proponer una propuesta de solución, fundamentada en la Hipótesis. Por último, se presenta la justificación del trabajo, la hipótesis a partir de la cual se plantea la solución, el objetivo general, los objetivos específicos, los alcances del trabajo.*

### 1.1. Breve Historia de la investigación Criminal

La primera disciplina precursora de la criminalística fue lo que en la actualidad se conoce como dactiloscopia, ciencia que estudia las huellas dactilares. La criminalística tal como la entendemos nace de la mano de la medicina forense, en torno al siglo XVII, cuando los médicos toman parte en los procedimientos judiciales. Antes de conocer el desarrollo y evolución de la criminalística debemos distinguir dos etapas.

Etapa equívoca: Eugene François Vidoq (1811).

Etapa científica: Alphonse Bertillon (1879), Juan Vucetich (1892), William Herschel, Francis Galton.

Algunos de los primeros usos prácticos de la investigación mediante las impresiones dactilares son acreditados a los chinos, quienes las aplicaban diariamente en sus negocios y empresas legales, mientras tanto el mundo occidental se encontraba en el período conocido como la edad oscura.

Kia Kung-Yen, historiador chino de la dinastía Tang, en sus escritos del año 650, mencionó la identificación mediante las impresiones dactilares, hacían mención del método anterior al uso de las impresiones consistentes en la utilización de placas de madera con muescas iguales recortadas en los mismos sitios de los lados las que conservaban las partes del contrato e igualadas dichas tablas se podía constatar la autenticidad o falsedad de los contratos de referencia.

Muchos años después, en 1575 surge otra ciencia precursora de la criminalística: la medicina legal, iniciada por el francés Ambrosio Paré y desarrollada por Paolo Sacchias en 1651.

En 1809 el célebre delincuente francés Vidocq fue incluido en las filas de la policía francesa y pronto se convirtió en el primer director de la Seguridad Nacional (Sûreté Nationale). Incluyó multitud de avances en el campo de la investigación criminal. A él se le atribuye el registro y creación de expedientes con las pesquisas de los casos y la introducción de los estudios de balística. Fue el primero en utilizar moldes para recoger huellas de la escena del crimen, definiendo la lofoscopia. Sus técnicas antropométricas tendrían gran repercusión. [CRI01]

Alfonso Bertillón creó en París el Servicio de Identificación Judicial en 1882, dado a conocer en 1885 y se adoptó de forma oficial en 1888. Este método antropométrico se basaba en el registro de las diferentes características óseas métricas y cromáticas de las personas mayores de 21 años en 11 diferentes partes del cuerpo.

El más ilustre y distinguido criminalista de todos los tiempos es Hanns Gross (1847-1915), se le considera el padre de la criminalística. A él se debe la generalización del término criminalística con el que se refería al «análisis sistemático de las huellas dejadas por el culpable».

En enero de 1920, el profesor Benjamín Martínez fundó en la Ciudad de México el gabinete de identificación y el laboratorio de criminalística en la entonces jefatura de Policía del Distrito.

## **1.2. Planteamiento del problema**

La seguridad a nivel nacional es un problema que ha afectado a México desde hace décadas, la ausencia de modernización en los métodos de investigación policiaca ha generado un aumento en los índices de criminalidad [pg1], una de las principales problemáticas que intenta abordar este trabajo está relacionado con la identificación de rostros de criminales, mediante la implementación de una base de datos; con la cual nos enfrentamos a otra problemática que es tener una base de datos de gran magnitud la cual genera un tamaño de archivo de millones de bytes requiriendo espacios de almacenamiento cada vez de mayor capacidad.

En México la herramienta biométrica más utilizada es la huella dactilar ya que es menos costosa de implementar, fácil de utilizar y con un marco de error mínimo, de igual manera tiene una gran desventaja este método biométrico, ya que la clonación de huellas dactilares ha incrementado notablemente [BBC], el reconocimiento facial tiene un muy bajo índice de clonación, ya que la clonación del reconocimiento facial no sucedería por su complejidad.

Lo que limita la implementación de algoritmos de reconocimiento facial en nuestro país es su alto valor(costo). Otra de las limitaciones es el gran almacenamiento que generan las bases de datos de la imagen es por ello por lo que, en este proyecto, se

pretende aplicar la transformada de wavelet para comprimir la base de datos del Reconocimiento Facial.

Es por ello por lo que es importante desarrollar un algoritmo eficiente y eficaz para que pronto sea empleado en diversos sectores de nuestro país.

### **1.3. Propuesta de Solución**

Para dar solución al problema planteado con anterioridad se implementará una base de datos comprimida, con los datos de cada individuo a través del uso de una transformada wavelet Haar se realizará una compresión en nuestra base de datos, por consiguiente se maximizarán los recursos del almacenamiento para la identificación de los individuos, se hará uso del algoritmo de reconocimiento facial Viola-Jones el cual ha demostrado ser uno de los algoritmos de reconocimiento facial con más eficiencia por lo anteriormente mencionado esta es una opción novedosa para recopilación de datos y estudio de criminalística.

### **1.4. Aportaciones**

- Reconocimiento facial más rápido y eficiente
- Compresión de la base de datos, de los individuos previamente registrados
- Optimización de los recursos de almacenamiento

### **1.5. Justificación**

El reconocimiento facial a nivel seguridad cuenta con muchas ventajas ya que el rostro humano cuenta con diversas características únicas e irrepetibles, las cuales pueden ser empleadas como identificadores clave.

Existen diferentes tipos de sistemas biométricos, uno de los más eficientes es el reconocimiento facial, ya que dicho sistema biométrico cuenta con una alta fiabilidad, facilidad de uso, prevención de ataques y una buena estabilidad.

Optimizar el tiempo de búsqueda y mejorar de una manera substancial los métodos tradicionales de investigación los cuales son totalmente ineficientes y anticuados por consecuencia reduciendo los índices de criminalidad presentes en el interior de la república mexicana.

Al implementar la transformada de wavelet Haar se realizará una compresión en nuestra base de datos, por consiguiente se maximizarán los recursos del almacenamiento, esto tendrá grandes ventajas para contar con bases de datos de gran capacidad y aun así poder seguir ingresando individuos al sistema.

## **1.6. Hipótesis**

El uso del sistema de reconocimiento facial es uno de los sistemas con más confiabilidad ya que el rostro humano tiene características muy específicas.

La implementación de un sistema de reconocimiento facial en el ámbito criminal tiene gran diversidad de ventajas ya que se podrán identificar individuos con mayor facilidad y precisión.

## **1.7. Objetivo General**

El objetivo de este trabajo es el de brindar una herramienta útil y eficaz para un control estadístico de la población basado en reconocimiento facial contando con una base de datos para el almacenaje y comparación de datos de vital importancia para futuras referencias o investigaciones en el ámbito de la investigación criminalística.

## **1.8. Objetivos Específicos**

- Determinar y aplicar el método de reconocimiento facial
- Lograr reconocimiento mediante capturas de imágenes
- Comparación de una imagen en específico en la base de datos previamente registrada
- Compresión de la base de datos, para lograr su optimización

## **1.9. Alcances del trabajo**

Con el presente trabajo se busca que dentro de unos años sirva como una herramienta para la investigación de la criminalidad en territorio mexicano por parte de corporaciones de investigación policiaca (FGR, Agencia de Investigación Criminal) con base en la información adquirida en investigaciones de criminales realizadas con anterioridad concentrándolas en un solo punto.

# Capítulo 2

## Estado del arte

*En algún lugar, alguna cosa increíble aguarda a ser descubierta.*

*Carl Sagan*

*El presente capítulo muestra los diferentes tipos de formato de imágenes, la compresión, los diferentes modos de compresión y los sistemas de colores.*

## 2.1. Imagen digital

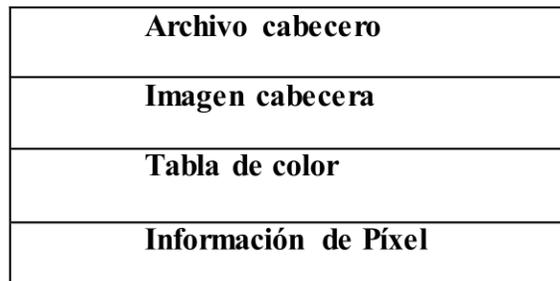
Una imagen puede ser definida como una función en dos dimensiones  $f(x, y)$ , donde  $x$  y  $y$  son coordenadas de un plano espacial y la amplitud de  $f$  de algún par de coordenada  $(x, y)$  es llamada intensidad o nivel de gris de la imagen en ese punto. Cuando  $(x, y)$  y la intensidad  $f$  son finitos y cantidades discretas podemos llamarle imagen digital. Dado que una imagen digital tiene elementos finitos, se puede decir que cada ubicación tiene un valor específico, estos elementos son llamados elementos de imagen o píxeles.

## 2.2. Formatos de Imágenes

### 2.2.1 Formato Windows BMP

BMP (Windows BitMap) es un formato propio del sistema operativo de Windows, de imagen de mapa de bits. Este formato es capaz de soportar imágenes con 1,4,8,16,24 y 32 bits por píxel. Sin embargo, la compresión BMP solo es útil con largos bloques de colores idénticos.

La estructura de un formato bmp es muy simple como se muestra en la siguiente figura.



**Figura 2.1:** Estructura de Archivo BMP [CIF00].

### 2.2.2. Archivo cabecera

El archivo cabecera especifica la locación de la información de píxel en el archivo. Cuando se decodifica un archivo BMP se deberá usar el campo `bfOffbits` para determinar el offset desde el inicio del archivo hasta donde la información de píxel empieza. La mayoría de las aplicaciones sitúa la información de Píxel inmediatamente seguido de la estructura `BITMAPINFOHEADER`.

**Tabla 2.1:** Estructura `BITMAPFILEHEADER`. [CIF01]

Campo	Tamaño en Bytes	Descripción
<code>bfType</code>	2	Contiene la palabra "BM" que identifica el tipo de archivo"
<code>bfSize</code>	4	Tamaño de archivo
<code>bfReserved1</code>	2	Sin utilizar
<code>bfReserved2</code>	2	Sin utilizar
<code>bfOffBits</code>	4	Offset para empezar la información de píxel

### 2.2.3. Imagen cabecera

Esta imagen se presenta en dos diferentes formatos, definidos como `BITMAPCOREHEADER` y `BITMAPINFOHEADER` esto último es el formato más común de Windows. Para determinar la estructura del tipo de imagen usada en particular se necesita examinar el tamaño del campo de estructura el cual son los 4 primeros bytes de ambas estructuras. A continuación, se muestra la estructura de ambos formatos.

**Tabla 2.2:** Estructura `BITMAPINFOHEADER`. [CIF02]

Tipo	Tamaño	Descripción
<code>b1Size</code>	4	Tamaño de cabecera (40)
<code>b1Width</code>	4	Ancho de Imagen
<code>b1Height</code>	4	Alto de Imagen
<code>b1Planes</code>	2	Debe ser uno
<code>b1BitCount</code>	2	Bits por píxel 1,4,8,16,24 o 32.
<code>b1Compression</code>	4	Tipo de compresión BI_RGB=0, BI_RLE8=1, BI_RLE4=2 o BI_BITFIELDS=3
<code>b1SizeImage</code>	4	Tamaño de Imagen puede ser cero si no está comprimido
<code>b1XPelsPerMeter</code>	4	Resolución de pixeles por metro
<code>b1YPelsPerMeter</code>	4	Resolución de pixeles por metro
<code>b1ClrUsed</code>	4	Número de entradas de mapa que están actualmente en uso.
<code>B1ClrImportant</code>	4	Número de colores significantes.

**Tabla 2.3:** Estructura `BITMAPCOREHEADER`. [CIF03]

Tipo	Tamaño	Descripción
<b>bcSize</b>	4	Tamaño de cabecera
<b>bcWidth</b>	2	Ancho de Imagen
bcHeight	2	Tamaño de Imagen
bcPlanes	2	Debe ser 1
bcBitCount	2	Cuenta de bit 1,4,8 o 24

#### 2.2.4. Paleta de Colores

Es el paso siguiente del archivo cabecera y puede estar representado en tres formatos. Los primeros dos son utilizados para hacer un mapeo del contenido de píxel a valores de color RGB cuando la cuenta de los bits es 1, 4 o 8.

Para los archivos BMP en Windows, la paleta consiste en un arreglo de  $2^{\text{cuenta de bit}}$  con la estructura mostrada en la siguiente tabla 2.4.

**Tabla 2.4:** Estructura RGB Doble. [CIF04]

Tipo	Tamaño	Descripción
rgbBlue	1	Valor de color azul
rgbGreen	1	Valor de a color Rojo
rgbRed	1	Valor de color verde
rgbReserved	1	Establecer como cero

Para utilizar un archivo BMP en formato OS/2 se utiliza un array de RGBTRIPLE como se muestra en la tabla 2.5.

**Tabla 2.5:** Estructura RGB Triple. [CIF05]

Tipo	Tamaño	Descripción
rgbtBlue	1	color azul
rgbtGreen	1	color rojo
rgbtRed	1	color verde

#### 2.2.5 Información de Píxel

La información continua después de la paleta de colores toma como base la estructura de BITMAPINFOHEADER o la estructura BITMAPCOREHEADER.

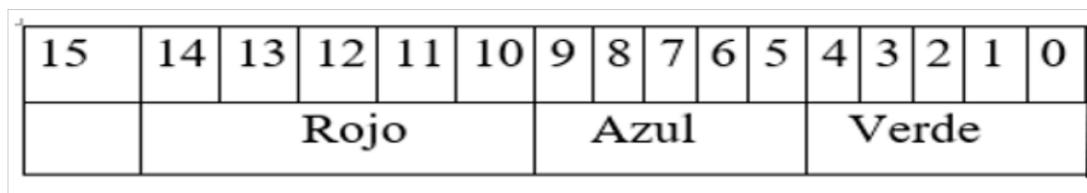
Las filas de píxel están ordenadas desde abajo hacia arriba, El número de filas está determinado por el tipo biHeight o bcHeight de la imagen cabecera y el tamaño es dado por biBitCount y biWidth.

El número de bytes por cada fila es calculado por la formula siguiente

$$\text{bytes per row} = \frac{\left(\frac{\text{width} \times \text{bitcount} + 7}{8}\right) + 3}{4} \quad (2.1)$$

El formato de la información de píxel depende del número de bits por píxel.

- 1 y 4 Bits por píxel: Cada información de bit esta subdividida en 8 o 2 tipos en los cuales los valores representan un índice de la paleta de colores.
- 8 bits por píxel: Cada fila de píxel es representada por 1 byte que es el índice de la paleta de colores.
- 16 bits por píxel: Cada píxel es representado por valores enteros de 2 bytes.  
El valor default por bit está representado en la siguiente figura 2.2.



**Figura 2.2:** Formato para 16 bits RGB. [POL1]

- 24 bits por Píxel: Cada píxel es representado por 3 bytes consecutivos que especifican los valores de rojo, verde y azul respectivamente. Hay que notar que este orden es contrario a los demás formatos de imágenes.
- 32 bits por Píxel: Cada píxel está representado por 4 bytes, los tres bytes de menor orden representan los 8 valores de bites para azul, verde y rojo en ese orden, el byte de mayor orden no se usa.

### 2.3. Compresión

El formato BMP soporta codificación de longitud de ejecución simple de 4 y 8 bits imágenes por píxel.

Una imagen de 4 bits es comprimida si el valor de biCompression de la estructura BITMAPINFOHEADER es BI\_RLE4 (=2), en cambio si nuestra imagen comprimida es de 8 bits el valor para compresión es BI\_RLE8 (=1).

La codificación de longitud de ejecución simple es una de las técnicas de compresión más simples que existen.

Este tipo de compresión es adecuado solo para imágenes que tienen muchos valores repetidos, como un dibujo para una caricatura.

Para muchos tipos de imágenes, la codificación de longitud de ejecución simple producirá un archivo de un tamaño muy grande (compresión negativa). El formato de compresión de datos depende del número de bits por píxel.

### **2.3.1. ¿Por qué usar este formato?**

Si el trabajo que se realiza para el desarrollo de software es leer y escribir archivos gráficos en el ambiente de desarrollo de Windows, esta es la mejor opción para utilizarlo para pruebas ya que es simple de implementar, fácil de depurar y puede ser visualizada sin ninguna herramienta especializada.

### **2.3.2. Formato JPEG**

El formato JPEG (Joint Photographic Experts Group) se ha convertido en el formato más común para almacenar información de imágenes fotográficas.

Es uno de los principales y más populares tonos continuos de compresión es el estándar JPEG. El cual define tres diferentes sistemas de codificación [Dip1].

1.- Sistema base con pérdida el cual está basado en DCT y es la adecuada por la mayoría de las aplicaciones con compresión.

2.- Sistema extendido de codificación si utiliza para una mayor compresión, mayor precisión o reconstrucción progresiva de aplicaciones.

3.- Sistema independiente y sin pérdidas para compresión reversible.

Para que nuestra compresión sea compatible con el estándar JPEG el producto o sistema debe incluir soporte para el sistema base; Este sistema base generalmente es llamado sistema base secuencial en donde las entradas y salidas está limitado a 8 bits mientras que los valores de DCT están restringidos a 11 bits. La compresión en si misma está constituida de 3 pasos: DCT, cuantización y asignación de código de longitud variable.

La imagen primero esta subdividida en bloques de pixeles de tamaño 8x8 los cuales están procesado de izquierda a derecha y de arriba hacia abajo. Como cada subimagen de 8x8 se encuentra entre sí, cada 64 pixeles el nivel cambia por la cantidad de  $2^n$  donde  $2^n$  es el número máximo de niveles de gris.

El gran éxito del formato JPEG es que para imágenes fotográficas obtenemos la mayor compresión de mapa de bits que cualquier otro formato, Una fotografía que requiere 1 Mb para ser almacenada en formato BMP puede ser comprimida hasta 50 KB con JPEG.

El formato JPEG no es perfecto ya que presenta algunas limitantes en algunas aplicaciones como por ejemplo no es bueno para utilizarlo en la compresión de textos o dibujos, así como también la edición de imágenes.

### 2.3.3. Modos de compresión

El estándar JPEG define cuatro modos de compresión: jerárquica, progresivo, secuencial y sin pérdidas.

En la siguiente figura se muestran los diferentes modos de codificación y su relación con los modos de compresión.

JPEG										
Secuencial				Progresivo				Sin pérdidas		Jerárquico
Huffman		Aritmético		Huffman		Aritmético		Sin pérdidas Original	JPE G-LS	
8 bits	12 bits	8 bits	12 bits	8 bits	2 bits	8 bits	12 bits			

**Figura 2.3:** Modos JPEG. [CIF06]

### 2.3.4. Progresivo

En el modo de compresión progresivo los componentes de la imagen están codificadas en escaneos múltiples. El contenido comprimido para cada componente es guardado en un mínimo de 2 y máximo de 826 escaneos. El primer escaneo crea una versión rugosa de la imagen mientras que en los escaneos subsecuentes se va refinando la imagen.

Las imágenes progresivas están hechas para estar siendo observadas mientras se está codificando.

Este tipo de compresión es útil cuando una imagen está siendo descargada a través de la red o usada en un buscador web debido a que nos da un panorama de que es lo que contiene la imagen después de que todos los datos han sido transmitidos.

### 2.3.5. Jerárquica

El modo de compresión jerárquica emplea un modo progresivo en el cual la imagen es dividida en un número de subimágenes esto se conoce con el nombre de frames. Un frame es una colección de uno o más escaneos. En el modo jerárquico el primer frame crea una versión de baja resolución de la imagen. En los siguientes frames se refina la imagen a través de aumentar la resolución de nuestra imagen.

La desventaja de esta compresión es su complejidad. El modo jerárquico requiere mucho más procesamiento que los demás modos de compresión y el uso de múltiples frames esto incrementa la cantidad de datos que deben ser transmitidos.

### 2.3.6. Sin pérdidas

El estándar definido como modo de compresión sin pérdidas siempre preserva la imagen original. Un modo sin pérdida nunca podrá comprimir tan bien una imagen como uno con pérdidas.

Debido a estas características no hay razón para utilizar este método de compresión.

Un nuevo método de compresión sin pérdida conocido como JPEG-LS se creó reemplazando totalmente al anterior modo de compresión sin pérdidas anterior.

## 2.4. Sistemas de colores

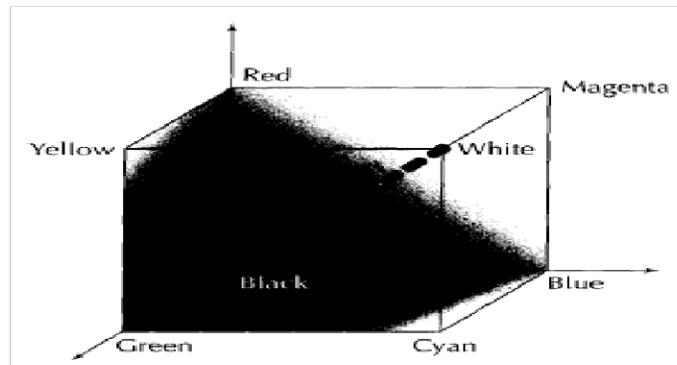
Existen muchas maneras para representar colores numéricamente uno de estos sistemas de para representar colores es llamado modelo de colores.

### 2.4.1. RGB

EL modelo más común de colores utilizadas en aplicaciones de computadora es también conocido como RGB(Red-Green-Blue).

En el modelo RGB, los colores están compuestos por tres valores que representa las intensidades relativas de rojo, verde y azul.

En la siguiente figura se muestra la relación de los colores en el modelo de colores RGB.



**Figura 2.4:** Modelo de color RGB. [CIF07]

El rango de color que puede ser representado por un modelo de color es conocido como espacio de color en la figura 1.1 el espacio de color está representado como un cubo.

En el ámbito matemático el color está representado con un rango de 0.0 a 1.0. En programación y formato de imágenes, valores enteros sin signo son los que son utilizados. El rango de estos valores para un componente de color es determinado por muestras de precisión el cual es el número de bits utilizados para representar el componente. Para imágenes fotográficas, 8 es el número más comúnmente utilizado para muestras de precisión sin embargo 1,2,4,12 y 16 también son comunes.

En Windows la muestra de precisión es siempre 8 bits, por lo que el sistema operativo reconoce 256 sombras diferentes de cada color primario.

Debido a esta cantidad limitada de 256 colores este sistema de colores está restringido a solo representar ciertos colores del espectro electromagnético.

Si representamos estos valores en una componente de imágenes 0 representara el negro y 255 representara el blanco.

Sin embargo, en muchas aplicaciones no es necesario representar tal cantidad de colores ya que solo se necesitan unos cientos de ellos y en algunas ocasiones solo unos pocos.

Dada la variedad de sistemas que están en uso, es útil tener un subconjunto de colores que reproduzcan fielmente a estos independientemente de las características de hardware.

Este subconjunto de colores es llamado colores de seguros RGB, o el set de todos los sistemas de colores seguros, en aplicaciones de internet son llamados colores seguros de red.

El estándar para todos los sistemas de colores seguros son 216, cada uno de estos 216 colores seguros está formado por los tres valores de RGB, aunque cada valor solo puede ser 0,51,102,153,204 o 255.

Estos valores también pueden ser representados en base Hexadecimal como se muestra en la tabla 2.6.

**Tabla 2.6:** Valores Validos de RGB. [DIP2]

Sistema numérico				Equivalente de colores		
<b>Hexadecimal</b>	<b>00</b>	<b>33</b>	<b>66</b>	99	CC	FF
<b>Decimal</b>	<b>0</b>	<b>51</b>	<b>102</b>	153	204	255

Donde se toman tres números para formar un color RGB, cada color seguro está formado por 3 o dos dígitos en base hexadecimal. Por ejemplo, el color rojo más brillante esta denominado como FF0000 donde R= FF (255) G=0 y B=0.

Y los valores 000000 y FFFFFFFF representa blanco y negro respectivamente.

### 2.4.2. Escala de grises

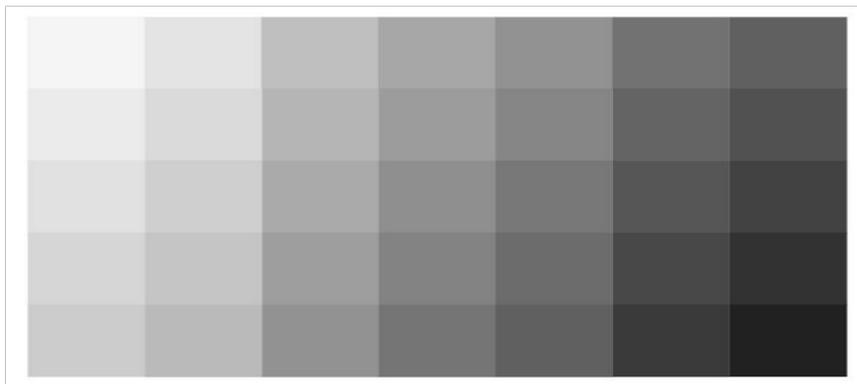
El rango de valores monocromáticos medidos de luz de negro a blanco es usualmente llamado escala de grises.

Algunos dispositivos de visualización como lo son impresoras lasares, no pueden mostrar el color en su totalidad en su lugar muestras sombras de gris.

También son conocidos como dispositivos de escalas de grises. Sombras de grises que pueden ser representadas por un componente simple en el rango 0 (negro) a  $2^{\text{Muestra de Precisión} - 1}$ .

El número de valores que abarcan las escalas de grises es variable dependiendo del uso que se le vaya a dar, aunque suele ser una cifra baja y sencilla de manejar, entre 3 y 10 valores, por cuestiones prácticas.

La escala de grises más simple es la de tres valores: blanco, gris y negro; seguida por la de **cinco** valores: blanco, gris claro, gris medio, gris oscuro y negro. Añadiendo a la escala de cinco valores un nuevo paso intermedio entre cada valor se obtiene la popular escala de 9 valores propuesta por Denman Ross en 1907.



**Figura 2.5:** Representación en escala de grises. [GRA1]

### 2.5. Comparativa entre sistemas biométricos

Existen diferentes tipos de sistemas biométricos los cuales tienen diferentes ventajas y desventajas. Por ello es importante el análisis de cada uno de ellos, para así saber su fiabilidad, facilidad de uso, prevención de ataques, aceptación y estabilidad.

**Tabla 2.7:** Diferentes características de los sistemas biométricos.

	Ojo(iris)	Ojo (retina)	Huellas dactilares	Vascular dedo	Vascular mano	Geometría de la mano	Escritura y firma	voz	Cara 2D	Cara 3D
Fiabilidad	Muy alta	Muy alta	Muy alta	Muy alta	Muy alta	Alta	Media	Alta	Media	Alta
Facilidad de uso	Media	Baja	Alta a	Muy alta	Muy alta	Alta	Alta	Alta	Alta	Alta
Prevención de ataques	Muy alta	Muy alta	Alta	Muy alta	Muy alta	Alta	Media	Media	Media	Alta
Aceptación	Media	Baja	Alta	Alta	Alta	Alta	Muy alta	Alta	Muy alta	Muy alta
Estabilidad	Alta	Alta	Alta	Alta	Alta	Media	Baja	Media	Media	Media

De acuerdo con la tabla 2.7 podemos observar que entre la comparativa mostrada entre los diversos sistemas biométricos, uno de los más eficientes es el reconocimiento facial ya que dicho sistema biométrico cuenta con una alta fiabilidad, facilidad de uso, prevención de ataques y una buena estabilidad.

De este modo en este proyecto se desarrollará el sistema biométrico mediante reconocimiento facial.

### **2.5.1. Detección de rostros**

La detección facial consiste en una tecnología de visión computarizada la cual determina los diferentes tipos de rostros humanos en imágenes o videos.

El algoritmo de detección de rostros presenta diversas dificultades debido a diversos factores que se presentan como:

- Variabilidad en la posición del rostro
- Presencia o ausencia de componentes estructurales (ejemplo: bigote)
- Expresión facial
- Oclusión (uso de lentes, gorra, etc.)
- Condiciones de ambiente
- Detectar el rostro humano es el primer paso en un sistema de reconocimiento facial. Por lo tanto, el algoritmo de reconocimiento facial debe ser capaz de identificar los rostros independientemente de los factores anteriormente mencionados.

## 2.5.2. Métodos de detección facial

Los métodos de detección facial se dividen en cuatro categorías

- a) Métodos basados en conocimiento: Codifican el conocimiento humano mediante distancias y posiciones entre las características humanas (ojos, nariz, labios).
- b) Métodos basados en características invariantes: Las características invariantes son aquellas las cuales no se pueden modificar, contrarias las características variantes, por ejemplo, al cambio de luz, pose o ubicación de la cámara. Este método funciona detectando uno de los componentes, construyendo un modelo estadístico para así verificar la existencia del rostro.
- c) Métodos basados en moldes(patrones): Es la relación entre la imagen de entrada y un patrón o molde previamente definido, cuyo objetivo es capturar características del rostro.
- d) Métodos basados en apariencia: Utilizan modelos obtenidos mediante entrenamiento de imágenes, tomando la imagen como un vector de características, a diferencia a los métodos basados en moldes, donde el patrón está definido por un “experto”, los patrones en este modelo son determinados por el aprendizaje obtenido en el entrenamiento de imágenes.

## 2.6. Detector de caras: Algoritmo de Viola Jones

Paul Viola y Michael Jones desarrollaron este algoritmo en 2001. Este sistema de detección de rostros representa un gran avance debido a la rapidez con la que cuenta para identificar rostros humanos, ya que se realiza una clasificación mediante características extraídas en una escala de grises, a diferencia de sus predecesores que lo realizaban píxel a píxel y en imágenes de color.

En el método de Viola Jones se definen los siguientes conceptos:

- **Integral de la imagen**, usada para la rápida detección de las características.
- Características de rectángulos, llamadas **características Haar**.
- **AdaBoost**, un método machine-learning, que consiste en reconocimiento de patrones y el estudio y construcción de algoritmos que aprenden y predicen el comportamiento de un conjunto de datos.
- Una cascada de decisión para combinar características de manera eficiente.

### 2.6.1. Integral de la imagen

Es una representación de la imagen original, permitiendo extraer las diferentes características de Haar.

Esta imagen es obtenida a partir de un algoritmo que genera la suma de los valores de un rectángulo dentro de una imagen.

La integral de una imagen se define como una matriz de igual tamaño que la matriz de la imagen original, que en la posición  $x, y$  contiene la suma de los píxeles contenidos arriba y a la izquierda del punto  $x, y$ , tal como se define en la fórmula:

$$\Pi(x, y) = \sum_{x' \leq x, y' \leq y} I(x', y') \quad (2.2)$$

Donde:

- $\Pi(x, y)$  es la integral de la imagen.
  - $I(x', y')$  es la imagen original.
- La integral de la imagen, en forma recursiva, es representada por:

$$\begin{aligned} S(x, y) &= S(x, y-1) + I(x, y) \\ \Pi(x, y) &= \Pi(x-1, y) + S(x, y) \end{aligned} \quad (2.3)$$

**Donde:**

$S(x, y)$  es la suma acumulada en fila.

$S(x, -1) = 0$  y  $\Pi(-1, y) = 0$

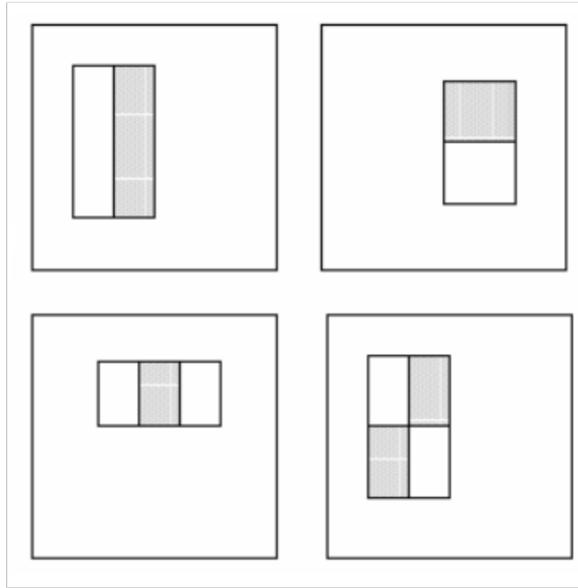
## 2.7. Características de Haar

Son descriptores que permiten obtener información de una zona en particular, las cuales se logran mediante operaciones aritméticas, la principal razón para usar esto es que permite gran eficiencia de cálculo.

La extracción de características es realizada aplicando filtros con base Haar.

En el algoritmo de Viola-Jones se usan tres características de Haar:

- Características de dos rectángulos: Es la diferencia entre la suma de píxeles de ambas regiones rectangulares.
- Características de tres rectángulos: Es la suma de los píxeles de los rectángulos exteriores menos la suma del rectángulo central.
- Características de cuatro rectángulos: Es la diferencia entre los pares diagonales de los rectángulos.



**Figura 2.6:** Ejemplo de características de rectángulo mostradas en relación con la ventana de dirección adjunta. [MA11]

## 2.8. Transformada Wavelet discreta

La transformada wavelet discreta es una transformación de la señal en donde se divide en dos tipos de sub-señales donde encontramos las aproximaciones y los detalles. Las aproximaciones son una copia de la señal, pero con una menor resolución, por lo tanto, contiene la mayor parte de la información o de energía de la señal original y los detalles almacenan la información referida a los cambios locales en la señal original.

La transformada discreta permite obtener las aproximaciones y detalles, con la información de mayor valor para poder realizar una compresión de señal descartando información. A partir de estos detalles se pueden realizar ciertos procesos como la detección de bordes, comparación, detección de cambios en la señal, etc.

Debido a que una sola transformada no puede resolver cualquier tipo de problema, y se debe de elegir una de entre todas las familias de wavelets que se adapte a las características de lo que se quiera estudiar.

La transformada wavelet discreta, puede adaptarse a uno o dos dimensiones; las dimensiones serán establecidas con las características de la señal. Para una señal de dos dimensiones, el mejor ejemplo es una imagen. Para el desarrollo de la transformada wavelet, hay tres técnicas: teoría de bancos de filtros, análisis multi resolución o análisis de tiempo-escala y codificaciones de sub-banda.

Si  $f(n) = f(x_0 + n\Delta_x)$  para  $x_0, n\Delta_x$  y  $n= 0,1,2,\dots,M-1$  así que los coeficientes de la serie de expansión wavelet  $f(x)$  son discretos y se convierten en los coeficientes de la secuencia de  $f(n)$ .

$$W_\varphi(j_0, k) = \frac{1}{\sqrt{M}} \sum f(n) \varphi_{j_0 k}(n) \quad (2.4)$$

$$W_\psi(j_0, k) = \frac{1}{\sqrt{M}} \sum f(n) \psi_{j_0 k}(n) \quad (2.5)$$

Donde  $\varphi_{j_0 k}(n)$  y  $\psi_{j_0 k}(n)$  son las funciones de las series wavelet de expansión muestreadas

### 2.9. Transformada Haar Wavelet

La transformada wavelet de Haar fue propuesta en 1909 por Alfred Haar y es reconocido como el primer wavelet conocido.

Para poder construir una transformada Haar tenemos que empezar construyendo un filtro de pasa bajas FIR filtro  $h = (h_0, h_1) = (\frac{1}{2}, \frac{1}{2})$ . El filtro toma a una secuencia de entrada  $c$  y produce vía  $h*x$  una secuencia  $y$  de valores promedios de  $x$  [VF01]. En particular se tiene.

$$y_n = \frac{1}{2}x_n + \frac{1}{2}x_{n-1} \quad (2.6)$$

El objetivo de esta tesis no es el de abordar extensivamente el desarrollo matemático por lo que se llega a la definición de transformada de wavelet.

$$W_N = \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \dots & 0 & 0 \\ \vdots & & & & \ddots & & \vdots \\ 0 & 0 & 0 & 0 & \dots & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \dots & 0 & 0 \\ \vdots & & & & \ddots & & \vdots \\ 0 & 0 & 0 & 0 & \dots & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix} \quad (2.7)$$

$$h = (h_0, h_1)$$

$$= \left( \frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2} \right)$$

Filtro Haar (2.8)

$$g = (g_0, g_1)$$

$$= \left( \frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2} \right)$$

Filtro Haar wavelet (2.9)

### 2.9.1. Transformada Haar de dos dimensiones como un producto de matrices

La TWD de Haar bidimensionales, son para señales que tienen dos parámetros, es un análisis y proceso similar al de una dimensión. Suponiendo una matriz A de tamaño N x N, donde N es par.

La transformada wavelet de Haar en dos dimensiones toma una matriz A y le asigna una nueva matriz que consta de 4 bloques.

Donde los coeficientes wavelet el bloque superior izquierda B es un 'blur' o aproximación de A, el bloque V representa diferencias verticales entre la matriz original y la aproximación, y los bloques inferiores H y D representa las diferencias horizontales y diagonales entre A y la aproximación [VF00]

Está definida como:

$$W_4 A W_4^T = 2 \begin{bmatrix} B & V \\ H & D \end{bmatrix} \quad (2.10)$$

Donde los coeficientes wavelet están definidos como:

$$B = \frac{1}{4} \begin{bmatrix} a_{11} + a_{12} + a_{21} + a_{22} & a_{13} + a_{14} + a_{23} + a_{24} \\ a_{31} + a_{32} + a_{41} + a_{42} & a_{33} + a_{34} + a_{43} + a_{44} \end{bmatrix} \quad (2.11)$$

$$V = \frac{1}{4} \begin{bmatrix} (a_{12} + a_{22}) - (a_{11} + a_{21}) & (a_{14} + a_{24}) - (a_{23} + a_{13}) \\ (a_{32} + a_{42}) - (a_{31} + a_{41}) & (a_{34} + a_{44}) - (a_{33} + a_{43}) \end{bmatrix} \quad (2.12)$$

$$H = \frac{1}{4} \begin{bmatrix} (a_{21} + a_{22}) - (a_{11} + a_{12}) & (a_{23} + a_{24}) - (a_{13} + a_{14}) \\ (a_{41} + a_{42}) - (a_{31} + a_{32}) & (a_{43} + a_{44}) - (a_{33} + a_{34}) \end{bmatrix} \quad (2.13)$$

$$D = \frac{1}{4} \begin{bmatrix} (a_{11} + a_{22}) - (a_{12} + a_{21}) & (a_{13} + a_{24}) - (a_{14} + a_{23}) \\ (a_{31} + a_{42}) - (a_{32} + a_{41}) & (a_{33} + a_{44}) - (a_{34} + a_{43}) \end{bmatrix} \quad (2.14)$$

Ejemplo

Se utilizará una matriz 4\*4 para este procedimiento

$$\begin{bmatrix} 16 & 20 & 14 & 8 \\ 8 & 10 & 0 & 12 \\ 11 & 12 & 9 & 20 \\ 9 & 10 & 15 & 10 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

Calculamos los coeficientes wavelet utilizando 2.10

$$W_4 A W_4^T = 2 \begin{bmatrix} B & V \\ H & D \end{bmatrix}$$

Para el coeficiente B utilizamos 2.11

$$B = \frac{1}{4} \begin{bmatrix} 16 + 20 + 8 + 10 & 14 + 8 + 0 + 12 \\ 11 + 12 + 9 + 10 & 9 + 20 + 15 + 10 \end{bmatrix}$$

$$B = \begin{bmatrix} \frac{54}{4} & \frac{34}{4} \\ \frac{42}{4} & \frac{54}{4} \end{bmatrix}$$

Para el coeficiente V utilizamos 2.12

$$V = \frac{1}{4} \begin{bmatrix} (20 + 10) - (16 + 8) & (8 + 12) - (0 + 14) \\ (12 + 10) - (11 + 9) & (20 + 10) - (9 + 15) \end{bmatrix}$$

$$V = \begin{bmatrix} \frac{6}{4} & \frac{6}{4} \\ \frac{2}{4} & \frac{6}{4} \end{bmatrix} = \begin{bmatrix} \frac{3}{2} & \frac{3}{2} \\ \frac{1}{2} & \frac{3}{2} \end{bmatrix}$$

Para el coeficiente H utilizamos 2.13

$$H = \frac{1}{4} \begin{bmatrix} (8+10) - (16+10) & (0+12) - (14+8) \\ (9+10) - (11+12) & (15+10) - (9+10) \end{bmatrix}$$

$$H = \begin{bmatrix} \frac{-18}{4} & \frac{-10}{4} \\ -1 & -1 \end{bmatrix}$$

Para el coeficiente D utilizamos 2.14

$$D = \frac{1}{4} \begin{bmatrix} (16+10) - (20+8) & (14+12) - (8+0) \\ (11+10) - (12+9) & (9+10) - (20+15) \end{bmatrix}$$

$$D = \begin{bmatrix} -\frac{2}{4} & \frac{18}{4} \\ 0 & \frac{-16}{4} \end{bmatrix} = \begin{bmatrix} -\frac{1}{2} & \frac{9}{2} \\ 0 & -4 \end{bmatrix}$$

Utilizando los coeficientes wavelet anteriormente calculados y aplicando 2.9

$$W_4 A W_4^T = 2 \begin{bmatrix} \begin{bmatrix} \frac{54}{4} & \frac{34}{4} \\ \frac{42}{4} & \frac{54}{4} \end{bmatrix} & \begin{bmatrix} \frac{3}{2} & \frac{3}{2} \\ \frac{1}{2} & \frac{3}{2} \end{bmatrix} \\ \begin{bmatrix} \frac{-18}{4} & \frac{-10}{4} \\ -1 & -1 \end{bmatrix} & \begin{bmatrix} \frac{1}{2} & \frac{9}{2} \\ -\frac{1}{2} & \frac{9}{2} \end{bmatrix} \end{bmatrix}$$

$$W_4 A W_4^T = \left[ \begin{array}{cc|cc} \frac{27}{1} & \frac{17}{1} & \frac{3}{1} & \frac{3}{1} \\ \frac{21}{1} & \frac{27}{1} & \frac{1}{1} & \frac{3}{1} \\ \frac{9}{1} & \frac{-5}{1} & \frac{-1}{1} & \frac{9}{1} \\ \frac{-2}{1} & \frac{-2}{1} & \frac{0}{1} & \frac{-8}{1} \end{array} \right]$$

## 2.10. Características útiles para la compresión de imágenes

La DCT tiene una buena capacidad de compactación de la energía al dominio transformado, es decir, que la transformada de coseno discreta consigue concentrar la mayor parte de la información en pocos coeficientes transformados tal y como muestra la imagen.

La transformación es independiente de los datos. El algoritmo aplicado no varía con los datos que recibe, como si sucede en otros algoritmos de compresión.

Hay fórmulas para el cálculo rápido del algoritmo, como podría ser la FFT para la DFT.

Produce pocos errores en los límites de los bloques imagen. La minimización de los errores a los bloques imagen permite reducir el efecto de bloque en las imágenes reconstruidas.

Tiene una interpretación frecuencial de los componentes transformados. La capacidad de interpretar los coeficientes en el punto de vista frecuencial permite aprovechar al máximo la capacidad de compresión.

DCT-1

$$f_j = \frac{1}{2}(x_0 + (-1)^j x_{n-1}) + \sum_{k=1}^{n-2} x_k \cos\left[\frac{\pi}{n-1}kj\right] \quad (2.15)$$

### 2.11. Histograma del Gradiente Orientado

Para detectar la presencia de una persona en una imagen existe una gran variedad de métodos, los cuales pueden ser clasificados dependiendo de diversas características. Actualmente uno de los métodos más destacados es la detección de personas utilizando histogramas del gradiente orientado (HOG por sus siglas en inglés: Histograms of Oriented Gradient).

Este método fue publicado en el año 2005, desde entonces a la fecha (27 de octubre del 2012) ha sido citado por alrededor de 6424 artículos y se han desarrollado alrededor de 106 versiones que han desarrollado pequeñas mejoras, pero en esencia el método no ha cambiado. En un principio el método HOG fue ideado como un detector de personas para imágenes en el espectro visual, es decir un método de un solo sensor.

El método HOG para la detección de personas busca representar una imagen utilizando como descriptor un conjunto de histogramas de las direcciones, estas direcciones discretizadas son llamadas canales de orientación y a partir de dicha representación se realiza una clasificación SVM para determinar que imágenes enmarca a una persona.

El detector de rostros HOG es bastante simple de entender, una de las principales razones para esto es que utiliza una función “global” para describir un rostro en lugar de una colección de características “locales”. En pocas palabras, esto significa que todo rostro está representado por un único vector de características, a diferencia de muchos vectores de características que representan partes más pequeñas de ese rostro.

El detector de rostros HOG usa una ventana de detección deslizante que se mueve alrededor de la imagen. En cada posición de la ventana del detector, se calcula un

descriptor HOG para la ventana de detección. Este descriptor se muestra luego al SVM entrenado, que lo clasifica como “rostro” o “no rostro”.

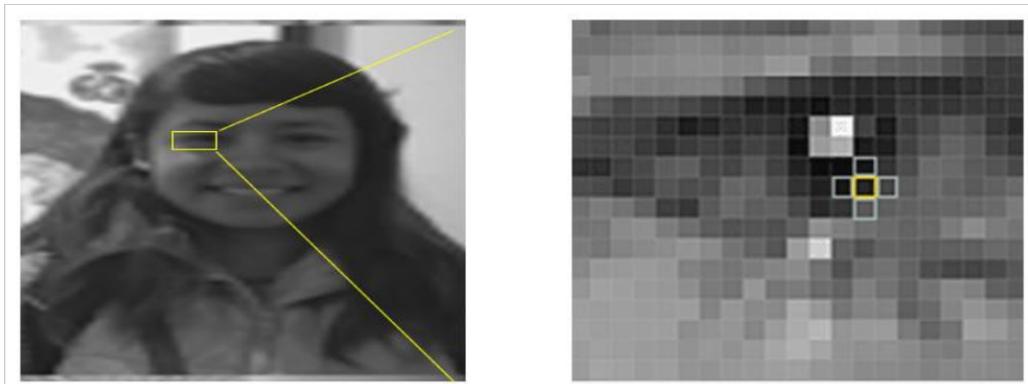
Para reconocer los rostros a diferentes escalas, la imagen se sub muestrea en varios tamaños. Se busca cada una de estas imágenes sub muestreadas.

Para encontrar rostros en una imagen, empezaremos haciendo que nuestra imagen sea en blanco y negro porque no necesitamos datos de color para encontrar rostros:



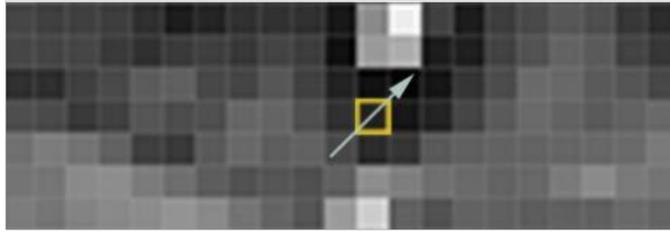
**Figura 2.7:** Ejemplo de imagen en blanco y negro para su análisis.

Se analizará cada píxel de la imagen de a uno por vez, para observar cada píxel, lo que se pretende ver son los píxeles que lo rodean directamente:



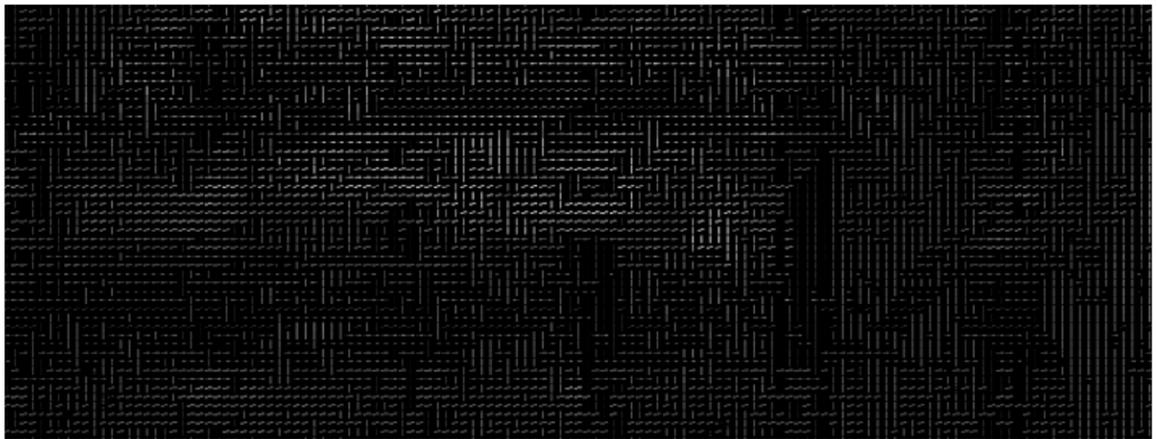
**Figura 2.8:** Análisis de píxeles.

El principal objetivo es determinar cuán oscuro, se compara el píxel actual con los píxeles que lo rodean directamente. Se procederá a dibujar una flecha que muestre en qué dirección la imagen se vuelve más oscura:



**Figura 2.9:** Orientación de flechas.

Este proceso se repite para cada píxel de la imagen, cada píxel es reemplazado por una flecha. Estas flechas se llaman gradientes y muestran el flujo de claro a oscuro en toda la imagen:



**Figura 2.10:** Análisis de brillo de las imágenes.

Se analizará la dirección en que cambia el brillo, tanto las imágenes realmente oscuras como las realmente brillantes terminarán con la misma representación exacta.

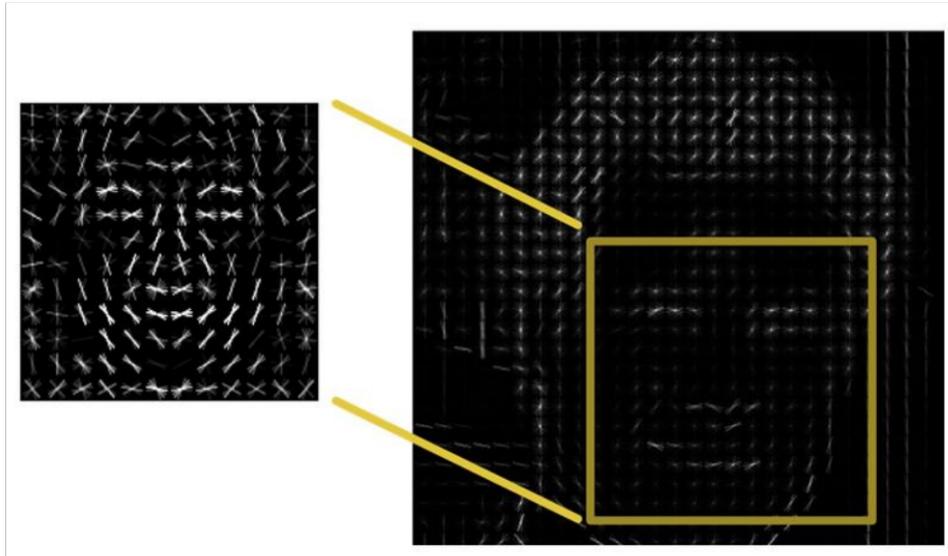
Pero al guardar el gradiente para cada píxel nos da demasiados detalles. Sería mejor si pudiéramos ver el flujo básico de luminosidad u oscuridad en un nivel más alto para poder ver el patrón básico de la imagen.

Para hacer esto, se divide la imagen en pequeños cuadrados de  $16 \times 16$  píxeles cada uno. En cada cuadro, vamos a contar cuántos puntos de gradientes en cada dirección principal (cuántos apuntan hacia arriba, cuántos apuntan hacia la derecha).

Luego reemplazaremos ese cuadrado en la imagen con las direcciones de flecha más fuertes.

El resultado final es que convertimos la imagen original en una representación muy simple que capta la estructura básica de una cara de una manera simple:

Para encontrar rostros en esta imagen de HOG, todo lo que tenemos que hacer es encontrar la parte de nuestra imagen que se parece más a un patrón de HOG conocido que se extrajo de un gran conjunto de otras caras de entrenamiento:



**Figura 2.11:** Detección de rostros.

## 2.12. Máquina de vectores de soporte (SVM)

Una máquina de vectores de soporte (SVM) es un algoritmo de aprendizaje supervisado que se puede emplear para clasificación binaria o regresión. Las máquinas de vectores de soporte son muy populares en aplicaciones como el procesamiento del lenguaje natural, el habla, el reconocimiento de imágenes y la visión artificial.

Una máquina de vectores de soporte construye un hiperplano óptimo en forma de superficie de decisión, de modo que el margen de separación entre las dos clases en los datos se amplía al máximo. Los vectores de soporte hacen referencia a un pequeño subconjunto de las observaciones de entrenamiento que se utilizan como soporte para la ubicación óptima de la superficie de decisión.

Las máquinas de vectores de soporte pertenecen a una clase de algoritmos de Machine Learning denominados métodos kernel y también se conocen como máquinas kernel.

El entrenamiento de una máquina de vectores de soporte consta de dos fases:

Transformar los predictores (datos de entrada) en un espacio de características altamente dimensional. En esta fase es suficiente con especificar el kernel; los datos

nunca se transforman explícitamente al espacio de características. Este proceso se conoce comúnmente como el truco kernel.

Resolver un problema de optimización cuadrática que se ajuste a un hiperplano óptimo para clasificar las características transformadas en dos clases. El número de características transformadas está determinado por el número de vectores de soporte.

Para construir la superficie de decisión solo se requieren los vectores de soporte seleccionados de los datos de entrenamiento. Una vez entrenados, el resto de los datos de entrenamiento son irrelevantes.

### **2.13. Códigos de salida de corrección de errores (ECOC)**

Códigos de salida de corrección de errores (ECOC), es un método de conjunto diseñado para problema de clasificación de varias clases. En el problema de clasificación de varias clases, la tarea es decidir una etiqueta de  $k > 2$  posibles opciones.

Por ejemplo, en la tarea de reconocimiento de dígitos, necesitamos asignar cada dígito escrito a mano a una de  $k = 10$  clases.

Algunos algoritmos, como el árbol de decisión, la red neuronal pueden manejar el problema de varias clases directamente.

# Capítulo 3

## Propuesta de solución

**Resumen:** En este capítulo se llevará a cabo la propuesta, así como el desarrollo del sistema de reconocimiento facial con compresión Haar Wavelet. Además, se incluirán los algoritmos utilizados paso a paso, así como la elaboración de interfaces para el uso del software utilizado.

La propuesta de solución que será empleada se basará en un sistema de reconocimiento basado en características de HOG, en la figura que se presenta a continuación se mostrará el funcionamiento general del prototipo.

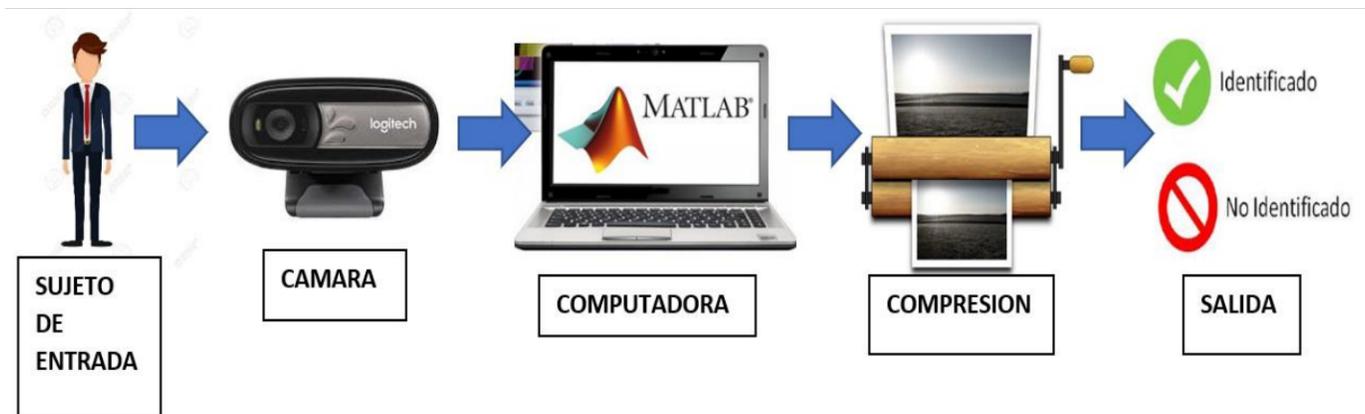


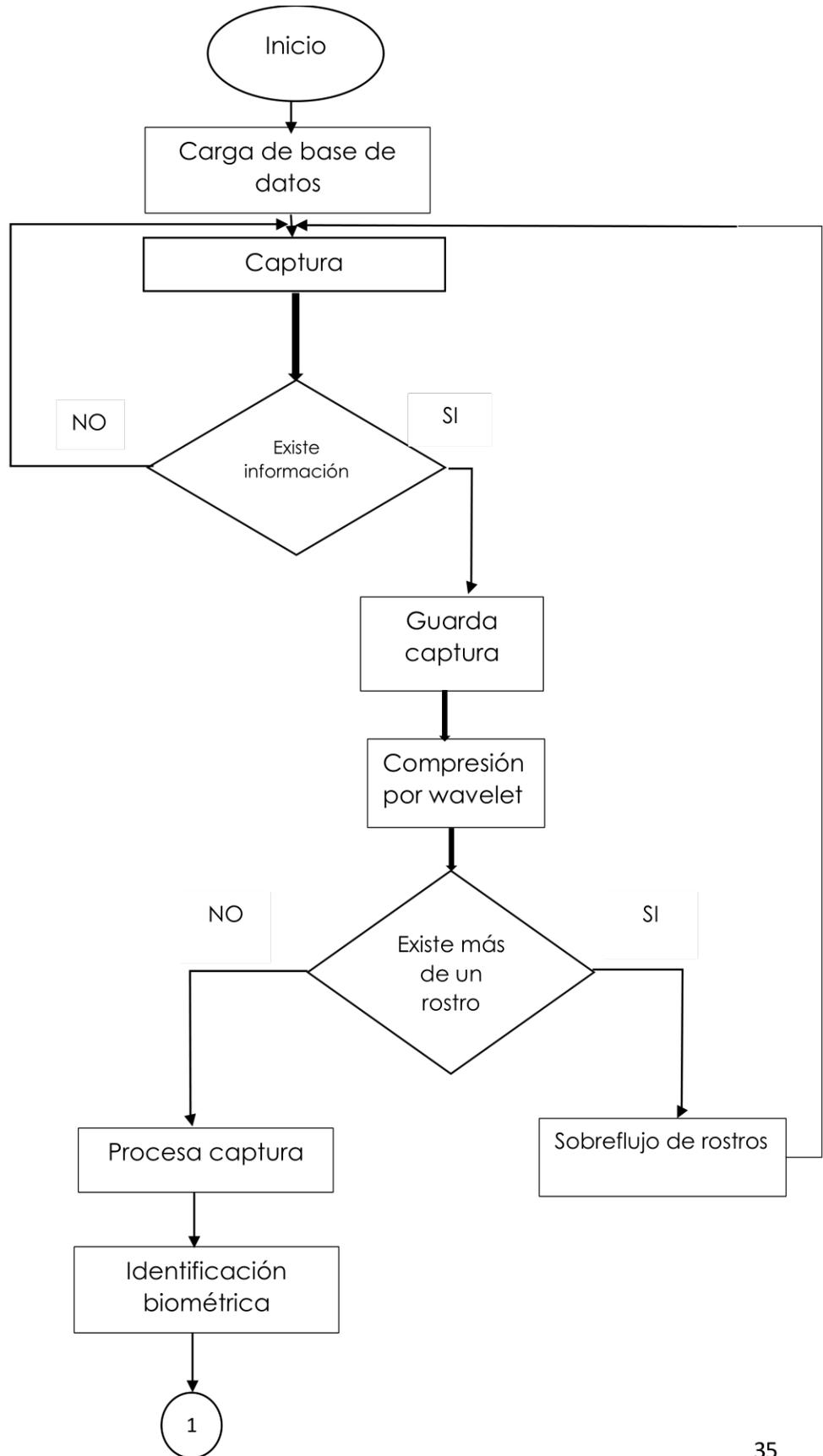
Figura 3.1: Funcionamiento general del prototipo.

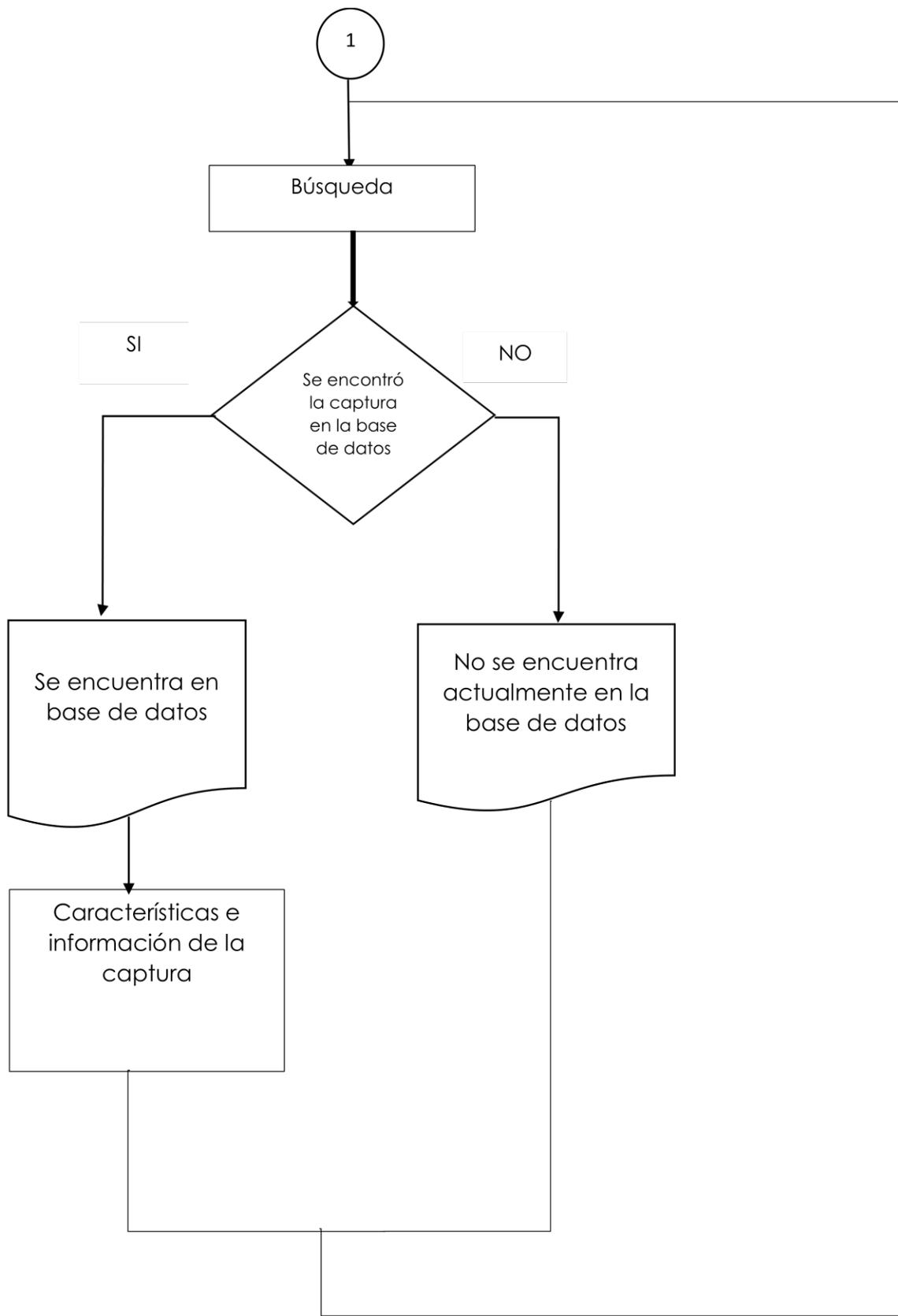
- **Sujeto de entrada:** Es el individuo al cual se le extraerán las características biométricas para poder determinar si está o no registrado en el sistema
- **Cámara:** Es el dispositivo que se encargara de capturar imágenes digitales para la comunicación de la computadora con el mundo real para que sean utilizados por el programa y la base de datos.
- **Computadora:** Es el dispositivo que compila y ejecuta las instrucciones del programa para aplicarlas visualmente
- **Compresión:** Este proceso se encargará de reducir el tamaño de almacenamiento de cada imagen
- **Salida:** Es la etapa que consiste en determinar si el sujeto se encuentra dentro de la base de datos arrojando resultados positivos o negativos según el caso.

### **3.1. Desarrollo del programa**

En este proyecto el procesamiento de las imágenes digitales será mediante el software de desarrollo MATLAB además de utilizar librerías del mismo software tal como *computer system toolbox* la cual nos proporcionará las herramientas y algoritmos usadas en la visión por computadora esencial para el desarrollo del proyecto.

El hardware utilizado para la obtención de las imágenes digitales será una webcam de la marca Logitech específicamente el modelo C170, las características de la computadora con la cual se realizaron las pruebas es un procesador Intel Celeron a 1.10 GHz, 4 GB de memoria RAM y sistema operativo Windows 10 así como MATLAB versión 2017b.





**Figura 3.2:** Diagrama de flujo.

### 3.1.1 Carga de base de datos

En la etapa inicial del programa se cargarán las características y el clasificador para los usuarios que serán registrados en el sistema.

Para poder realizar este procedimiento se genera un objeto que contendrá el conjunto de datos biométricos de cada uno de los sujetos, cada conjunto será identificado con una descripción (Description), el número de imágenes en formato bmp contenidas (**Count**) que representan las 20 imágenes almacenadas en la base de datos además de una descripción y la ubicación de cada imagen (**Image Location**).

La instrucción que nos arroja todas las características anteriormente dichas es la instrucción **imageSet** la cual requiere 2 parámetros de entrada uno es la carpeta donde se encuentran las imágenes, el segundo parámetro es la instrucción '**recursive**' para que se incluya las imágenes que se encuentran en las subcarpetas dentro de la carpeta '**BaseUsuarios**' ya que cada usuario tendrá una carpeta individual, este objeto será nombrado detector.

```
detector = imageSet('fotos','recursive');
```



Description	'fotos'
ImageLocation	1x20 cell
Count	20

Figura 3.3: Contenido del objeto detector

Para poder realizar las pruebas de entrenamiento con las 20 imágenes almacenadas es necesario realizar particiones de cada una de ellas, por ejemplo, se utilizarán 18 imágenes de la base de datos y las 2 últimas se utilizarán para realizar las pruebas.

La instrucción que nos auxiliaría para dividir las imágenes es la denominada como **partition** en entrenamiento y prueba utilizando el objeto detector.

```
[entre,prub]= partition(detector,[0.9 0.1]);
```

El porcentaje que toma la instrucción es de un 90% de imágenes para el entrenamiento y un 10% serán utilizadas para realizar las pruebas, Posteriormente se mostrarán los resultados obtenidos.

Una vez realizado esto será necesario crear un clasificador extrayendo las características biométricas por cada imagen, las características serán las del histograma de gradiente orientado o **HOG** en lengua inglesa.

Debido a que necesitamos obtener las características de cada imagen una por una se hará uso de un ciclo **for** generando bucles anidados, el bucle interno extraerá las características utilizando la instrucción **extractHOGFeatures** perteneciente a la toolbox de MATLAB dándole a cada imagen una etiqueta para poder darle una particularidad a cada una y por consiguiente llegar a diferenciarlas, el bucle externo llevara el conteo de los usuarios en nuestra base de datos y guarda en Des\_usuario la descripción de cada usuario.

Las características extraídas son guardadas en una matriz con el nombre Carac\_entrenamiento, Cont\_caracteristica ayuda a cambiar de fila al terminar con una imagen.

```
for i=1:size(entre,2)
for j=1 :entre(i).Count
Feat_entrenamiento(Cont_caracteristica,:) = extraHOGFeatures(read(entre(i),j));
Etiqueta(Cont_caracteristica) = entrenamiento(i).Description;
Cont_caracteristica= Cont_caracteristica + 1;
end
Des_usuario {i}= entrenamiento(i).Description;
end
```

El tamaño de la matriz será dado el número de filas esta dado por las fotos de entrenamiento (18) multiplicado por el número de sujetos en la base de datos (8) y las columnas son el número de características de HOG entregadas por la función para extraerlas, por lo que la matriz tendrá un tamaño de (144 x 4680).

Ya que no es posible modelar matemáticamente a cada sujeto debido a las características de memoria y velocidad del dispositivo utilizado se recurre a hacer un modelado por medio de un clasificador ECOC ya que permite corregir los errores causados en los algoritmos de aprendizaje.

Una vez teniendo las características y etiquetas de las imágenes del entrenamiento, se procede a utilizar la instrucción **fitcecoc** para generar un clasificador que nos permita diferenciar a cada usuario registrado en la base de datos, este clasificador

entrega 40 clases cada una con el nombre de la etiqueta enviada (Etiqueta\_env). En este caso se recibe un modelo multiclase ECOC utilizando modelos de vectores binarios SVM  $k(K-1)/2$  usando codificación 1 vs 1. Donde K representa el número de etiquetas de clase única. Este modelo será almacenado en la variable clasificador para utilizarlos posteriormente.

Con este modelo se puede determinar si el sujeto esta registrado o no en la base de datos. Para realizar se utiliza la función predict, para hacer la predicción de la imagen ingresada, para utilizar comparaciones es necesario hacer un preprocesamiento de la imagen, una vez que se tiene el modelo clasificación se utiliza el algoritmo de reconocimiento facial detallado a continuación.

### 3.1.2. Etapa de captura

Para este proceso se utilizará una cámara digital USB o webcam con resolución VGA (640 x 480 pixeles) esta resolución cumple con los requisitos mínimos de resolución por lo que no se necesita de una cámara de un gran costo o alta resolución, además se utiliza escala de grises, así como un formato de imágenes BMP para el almacenamiento de imágenes.

La cámara utilizada es una Logitech C170 la cual contiene todas las características y requerimientos para la captura de imagen (Apendice B).

Para iniciar se crea un objeto de video como el que se detalla a continuación:

```
vid= videoinput('winvideo',2);
```

Donde vid representa el objeto de video de entrada generado por videoinput (para poder utilizar este comando se necesita descargar una librería de Matlab), con los argumentos de entrada el adaptador de video de la computadora **winvideo** seguido de la webcam conectada, en este caso es el numero 2 debido a que las características de hardware ya contenía una webcam con anterioridad por último se selecciona el formato de la imagen BMP en el caso de la resolución MATLAB tomara como defecto una resolución VGA 640 x 480 pixeles.

Teniendo las configuraciones para poder utilizar la cámara procedemos a capturar imágenes mediante la función **getsnapshot**, la cual guardará una imagen en un espacio de color RGB para la foto de muestra, esta foto posteriormente será convertida a escala de grises ya que con ella se puede trabajar con más facilidad y con mejor desempeño.

### 3.1.3. Etapa de compresión

Una vez realizada la captura y almacenaje de la imagen se procede a realizar la compresión mediante la transformada wavelet de Haar debido a que por su simplicidad y sus características de un gran rango de compresión es la ideal para esta etapa.

Como primer paso se necesita leer la imagen que se le realizara la compresión, esto se lleva a cabo instrucción **imread**, su argumento de entrada será el nombre de la imagen con la extensión del formato (.bmp), el valor que arroje esta instrucción será almacenada en **mpbit**.

```
mpbit=imread('imagen40.bmp');
```

Ya concluido el proceso de lectura, generamos una wavelet de Haar bidimensional (utilizada para imágenes digitales) esto se logra mediante la instrucción **dwt2** con argumentos de entrada, matriz de la imagen a comprimir (**mpbit**) y tipo de transformada utilizada (**haar**).

Retornando los coeficientes wavelet [a, h, v y d] (para mayor referencia ver pág. 26). Renombrados como [ca, ch, cv y cd] para futuras referencias.

Por consiguiente, se tiene

```
[ca,ch,cv,cd]= dwt2(mpbit,'haar');
```

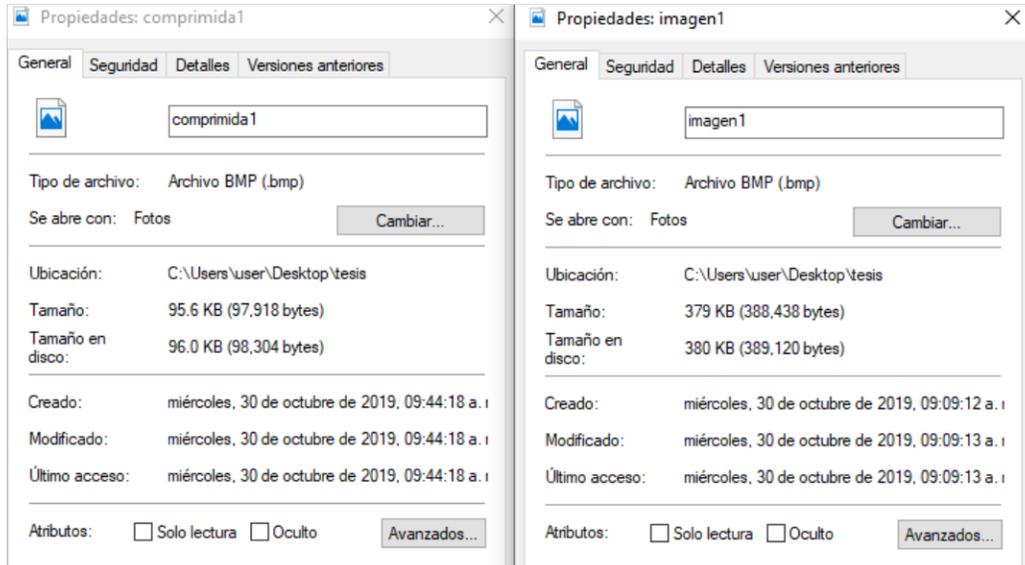
Para poder lograr la compresión deseada solo se utilizará el coeficiente a que se obtuvo ya que este coeficiente nos da un panorama suficiente de la imagen utilizada; Debido a que se necesita trabajar en una base comprensible para el ser humano transformamos el coeficiente en números enteros en formato de 8 bits lográndolo gracias a la instrucción **uint8** almacenándolo en **mbca**.

```
mbca=uint8(ca);
```

El último paso será guardar la imagen, ya comprimida, nos auxiliamos con la instrucción **imwrite** para conseguirlo con valores de entrada de valores en formato de 8 bits (mbca) y el nuevo nombre que recibirá la imagen comprimida (xxxxxx.bmp).

```
imwrite(mbca,'comprimida40.bmp');
```

En la imagen mostrada a continuación se muestra el grado de compresión obtenida



a) Imagen Comprimida

b) Imagen sin compresión

**Figura 3.4:** Comparación de almacenamiento entre imagen comprimida(a) vs imagen sin compresión(b).

Como se puede observar en la figura 3.3 a esta ocupa un espacio de almacenamiento de 96 Kb al contrario de la imagen b que ocupa 380 kb por lo que el porcentaje de compresión obtenido es de un 74.68% por lo que la compresión es comprobada.

### 3.1.4. Detección de rostros

En la siguiente fase se usarán las imágenes anteriormente comprimidas la compresión de detección de rostros se hace mediante la función **visión.CascadeObjectDetector()** y la función **step**.

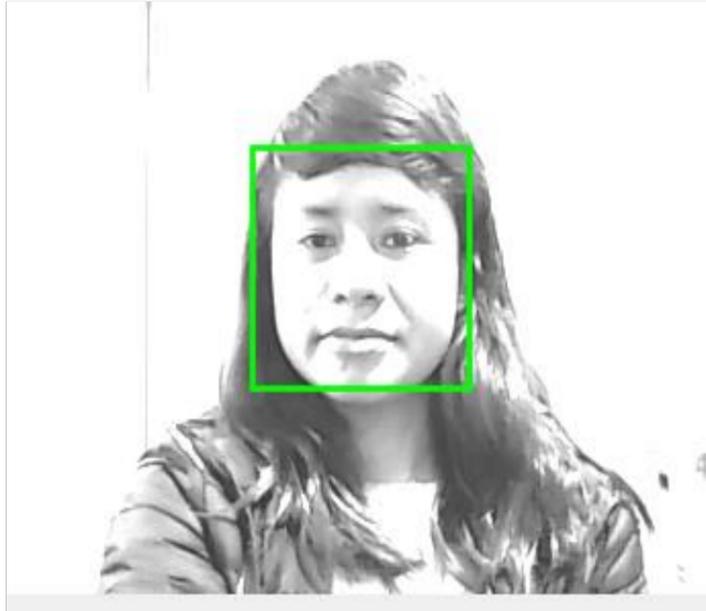
La función **visión.CascadeObjectDetector()** regresa el modelo de detección de rostros para una imagen digital mediante el algoritmo de Viola-Jones, la función **step** se utiliza para encontrar los rostros.

```
detec= vision.CascadeObjectDetector();
```

```
ros= step(detec,i);
```

Donde **ros** es la variable que almacena los datos del rostro detectado en un vector que contiene  $x_{min}$ ,  $y_{min}$  el ancho y el alto del rostro detectado en pixeles con las coordenadas  $x_{min}$ ,  $y_{min}$  se tiene el lazo superior izquierdo para formar el cuadrado que delimita el rostro detectado con el alto y ancho dados por la misma función.

En caso de haber más de un rostro, se tomara una forma de matriz  $N \times 4$  donde el número de filas es el número de rostros detectados, por consiguiente, las columnas alojan las coordenadas de los rostros.



**Figura 3.5:** Detección por Viola Jones

El sistema seguirá capturando cada 3 segundos verificando si hay rostros, la captura almacenada en memoria volverá a realizar el proceso anteriormente visto para la detección de rostros.

### **3.2. Desarrollo Interfaz gráfica**

MATLAB es un programa que facilita la creación de Interfases Gráficas de Usuario (GUI), en la interfaz existen algoritmos tales como el registro de usuarios, la información de usuarios en base de datos, sistema de seguridad de un cajero y reconocimiento de sujetos. Esta interfaz funciona de la siguiente manera.

En el menú principal se observan 4 botones, los cuales contienen todas las características utilizables, encontramos: registro, sistema de seguridad, detección de rostros y base de datos.



**Figura 3.6:** Menú Principal Interfaz Grafica

### **Registro**

Este botón es utilizado para registrar a los usuarios en la base de usuarios, al interactuar con este botón se visualiza la siguiente figura



**Figura 3.7:** Menú Registro

Ya dentro del apartado registro se muestra una ventana superior del lado izquierdo el cual representa el campo visual de la cámara, en la pantalla negra inferior se mostrará el resultado de la fotografía, del lado derecho encontramos 4 botones denominados como “Encender Cámara”, “Captura”, “Siguiente” y “Cancelar”.

## Encender Cámara

Como su propio nombre lo indica este botón sirve para encender y crear un objeto de video, el código utilizado fue el siguiente.

```
global vid;
axes(handles.pantalla1);
vid= videoinput('winvideo',2);
vid.ReturnedColorSpace='grayscale';
vidRes= get(vid,'VideoResolution');
nBands= get(vid,'NumberofBands');
hImage = image(zeros(vidRes(2),vidRes(1),nBands));
preview(vid,hImage);
```

## Captura

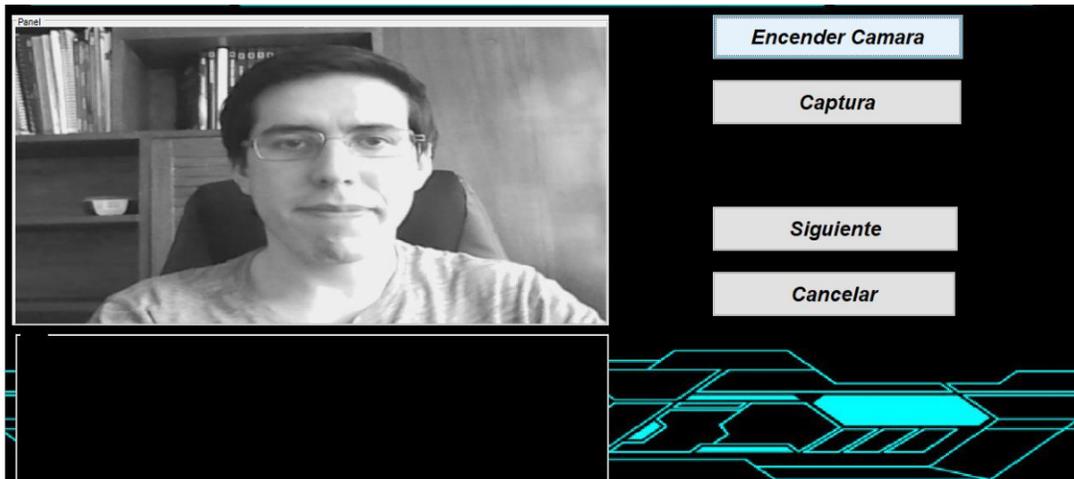
Este botón es utilizado para tomar las fotografías para la base de usuarios el cual con la función `getsnapshot()` permite tomar un fotograma de lo que se encuentra en ese momento, además con la función `imresize` se reajusta el tamaño de la fotografía para poder detectar el rostro de una forma más precisa y adecuada.

La captura de la fotografía se hace mediante la camera Logitech C170, en escala de grises y con reajuste.

Utilizando el siguiente programa:

```
foto= getsnapshot(vid);
vid.ReturnedColorSpace='grayscale';
a=imresize(foto,0.7);
```

Este código genera lo que se muestra en la figura 3.8



**Figura 3.8:** Botón Captura

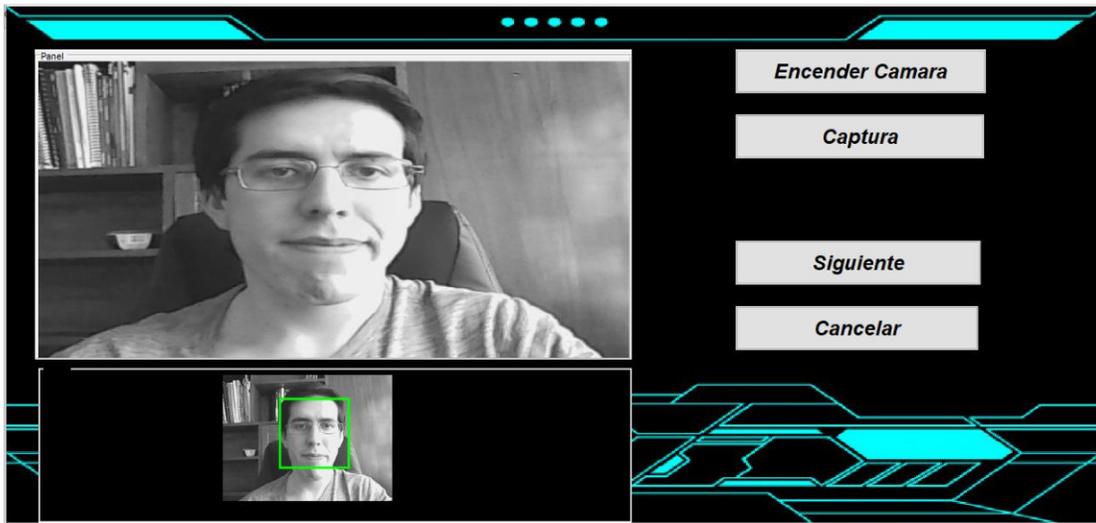
Una vez capturada la imagen se realiza la detección de rostros con el algoritmo de Viola-Jones, el cual encerrará el rostro mostrándola en la pantalla inferior si fue correcta se encerrará en un cuadro y será tomada como correcta como se muestra en la figura 3.9

Para la detección de rostros se utilizó el siguiente código:

```

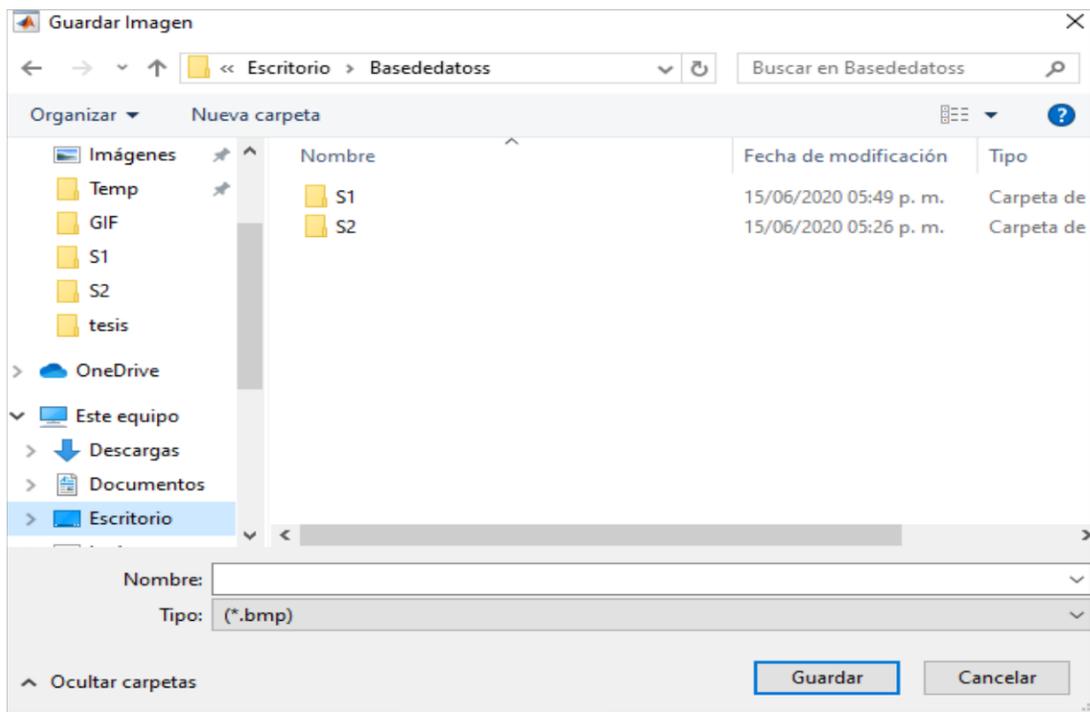
detec= vision.CascadeObjectDetector();
ros= step(detec,i);
imshow(i);
rectangle ('Position', ros (1, :),'edgecolor','g','linewidth',2);
formato= (*.bmp');
[nomb,ruta]= uiputfile(formato,'Guardar Imagen');
if nomb==0
return
end
fname=fullfile(ruta,nomb);
imwrite(a,fname);

```



**Figura 3.9:** Botón captura con detección de rostros.

Posteriormente se pedirá al usuario que le asigne un nombre y la guarde en la carpeta de base de datos en una carpeta vacía como se muestra en la figura 3.10.



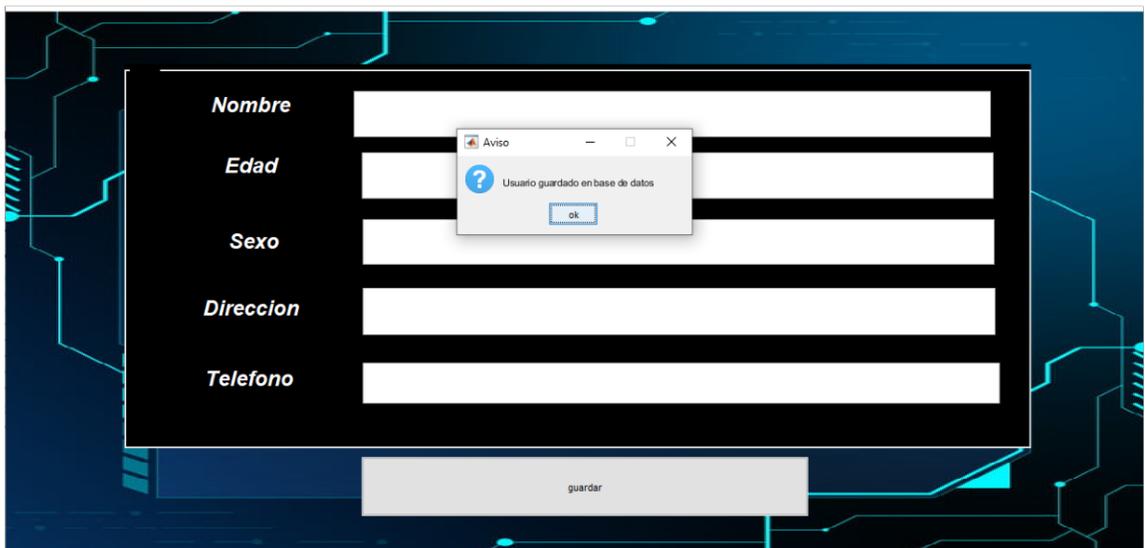
**Figura 3.10:** Guardado de Fotografía en Base de Datos

## Siguiente

Una vez realizada la captura de la fotografía, el botón “Siguiente” se encargará de borrar lo que se encuentre en pantalla y dejará al usuario tomar 10 fotografías, en cuanto se llegue a ese número este botón desaparecerá, dando lugar a la aparición de registro de usuario.

## Registro

Se le pedirá al usuario que, de su nombre, edad, sexo, dirección y teléfono, una vez ingresados estos datos, presionamos el botón guardar, si los datos ingresados fueron correctamente guardados se mostrara un cuadro de dialogo como el de la figura 3.10



**Figura 3.11:** Registro exitoso de usuario

## Cancelar

La función de este botón es regresar al menú principal y cerrar la cámara

Utilizando el siguiente código:

```
close(interfaz3);
```

```
interfaz1;
```

El siguiente botón que puede ser visualizado dentro de la interfaz principal es el de sistema de seguridad.

### **Sistema de Seguridad**

Al seleccionar este botón se muestra la interfaz siguiente de la figura 3.11



**Figura 3.12:** Interfaz Sistema de Seguridad

#### **Encender Cámara**

Este botón sirve como su mismo encender la cámara y crear un objeto de video.

#### **Iniciar**

El botón iniciar tiene la función de ver cuántos usuarios están conectados en ese momento, existen dos posibilidades:

- Un usuario
- Dos o más usuarios

#### **Caso a**

En el caso en donde solo exista un usuario el sistema marcara como seguro el ingreso del usuario al cajero mostrándose como en la figura 3.13.



Figura 3.13: Sistema de seguridad con un usuario.

Una vez que el sistema que es seguro seguir, se le indica al usuario que presione el botón continuar.

### Continuar

Con este botón el usuario podrá ingresar de una manera segura al cajero como se muestra en la figura 3.14.



Figura 3.14 Acceso del usuario con seguridad

## Salir

Con este botón regresamos al menú de sistema de seguridad

## Caso b

En el caso de que exista más de un usuario el sistema considerara que existe una falla en la seguridad por lo que enviara una señal de alerta en forma audible y visual como el de la figura 3.14.



**Figura 3.15:** Intento de Ingreso con falla de seguridad

En este caso el sistema no permitirá el acceso por lo que el usuario tendrá que cancelar la operación.

## Cancelar

Regresara al Menú Principal

Ahora en el caso de detección de rostros se tiene

## Detección de Rostros

Al presionar este botón nos enviara una interfaz como la de la figura 3.16.



Figura 3.16: Interfaz Búsqueda de usuarios.

Aquí se muestran 3 botones, ingresar foto, tomar captura y cancelar, el cual su funcionamiento a detalle será explicado a continuación.

### Ingresar foto

En esta opción podremos reconocer a un usuario con una imagen que se tenga previamente, nos pedirá que seleccionemos la imagen que deseamos que sea comparada con la base de datos, tal como se muestra en la figura 3.16.

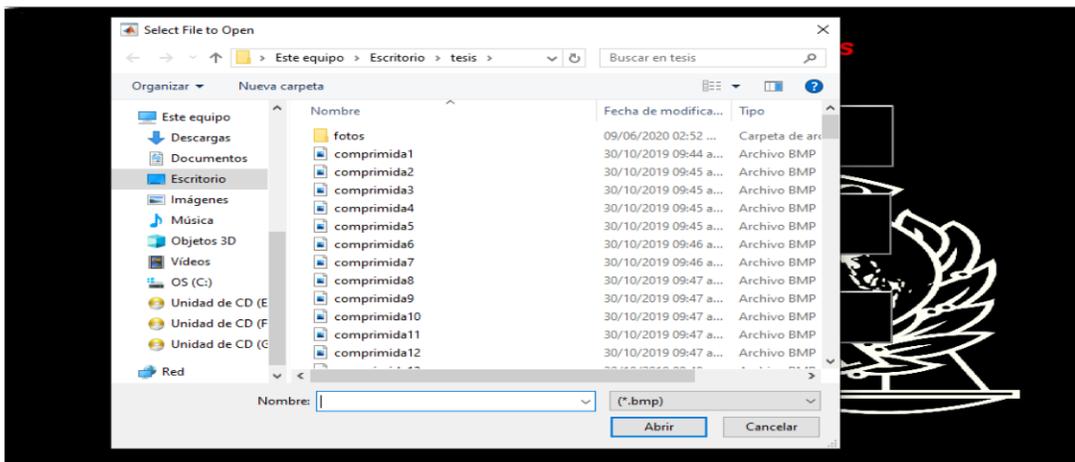


Figura 3.17: Selección de Imagen

Una vez que seleccionemos la imagen deseada nos dirigirá a una interfaz secundaria como el de la figura 3.18.

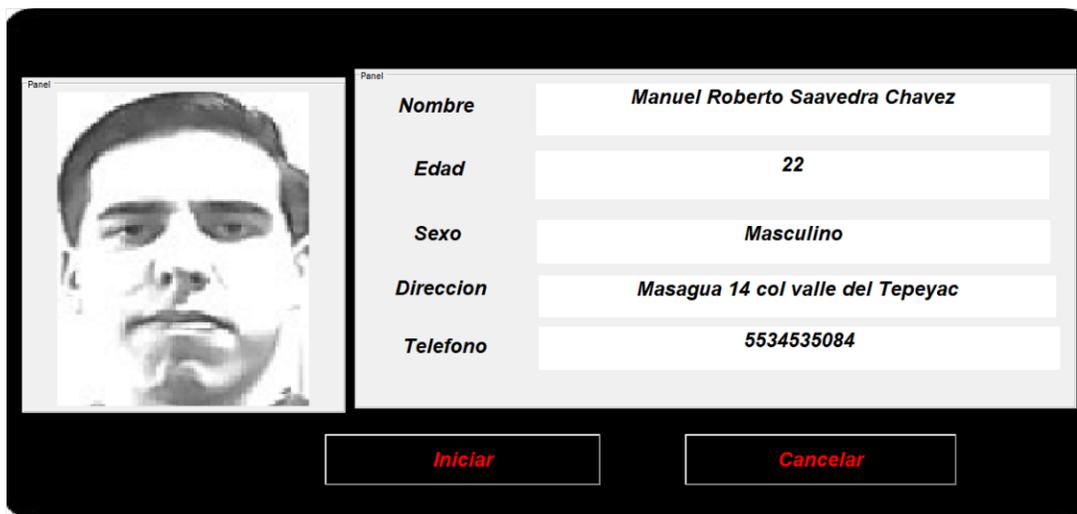


**Figura 3.18:** Interfaz secundaria Ingresar foto

En esta interfaz encontramos dos cuadros que mostraran datos del sujeto, una mostrara la fotografía y el del lado derecho mostrara sus datos personales, adicionalmente se visualizan dos botones Iniciar y continuar.

### **Iniciar**

Este botón es el que comparara la imagen con la que tenga registrada en la base de datos una vez pulsado se mostraran los datos y fotografía del sujeto a identificar como el de la figura 3.19



**Figura 3.19:** Identificación del sujeto mediante imagen.

### **Cancelar**

Se utiliza para regresar al menú de búsqueda de usuarios.

Por consiguiente, se describe el botón tomar captura

### Tomar captura

Este botón es utilizado cuando no se cuenta con una fotografía del sujeto del usuario y se quiere tomar la fotografía en tiempo real, al pulsar el botón se activará la cámara y aparecerán dos botones captura y cancelar como en la figura 3.19.



Figura 3.20: Interfaz tomar captura.

### Captura

Con este botón se toma la fotografía para poder realizar la búsqueda del sujeto, la fotografía será tomada en escala de grises y con dimensiones 127x158 pixeles, una vez que sea tomada con éxito será detectada el rostro encerrado con un cuadro verde como el de la figura 3.20.



Figura 3.21: Fotografía tomada con éxito para la búsqueda en base de datos.

Presionamos el botón continuar para pasar a la siguiente interfaz

### Continuar

Al presionar este botón se nos muestra el resultado de nuestra fotografía y el botón iniciar como el de la figura 3.22.



Figura 3.22. Botón Continuar.

### Iniciar

El uso de este botón es encontrar en la base de datos al usuario con una puntuación más cercana al de nuestra imagen, una vez que lo encuentra nos muestra sus datos y fotografía, como en la figura 3.22



**Figura 3.23:** Identificación de usuario mediante captura de fotografía en tiempo real.

### **Cancelar**

Regresamos al menú principal

El ultimo botón que se encuentra en la interfaz principal es de la base de datos el cual se detallara su funcionamiento a continuación.

### **Base de Datos**

Al activar este botón nos dará arrojará todos los usuarios que se encuentren registrados en ese momento en la base de datos arrojando la siguiente interfaz (ver figura 3.23).



**Figura 3.24:** Interfaz base de usuarios.

En la interfaz de base de usuarios nos presenta el botón iniciar el cual tiene la siguiente característica.

### **Iniciar**

Al interactuar con este botón nos muestra el 1er usuario registrado en la base de datos, su fotografía en la parte izquierda y su información personal en la parte derecha como se muestra en la figura 3.24.



**Figura 3.25:** 1er Usuario registrado en base de datos.

Además, se observa que aparecen 2 botones, anterior y siguiente

#### **Anterior**

Este botón es usado para retroceder al usuario anterior de la base de datos que haya sido registrado con anterioridad, en donde su código es el siguiente.

```
handles.counter = handles.counter - 1;
cla(handles.axes2,'reset');
files = dir('C:\Users\user\Desktop\Basededatoss\**\*.mat');
fn=fullfile(files(handles.counter).folder,files(handles.counter).name);
m1=load(fn);
axes(handles.axes2);
filesim= dir('C:\Users\user\Desktop\Basededatoss\**\recortada1.bmp');
im=fullfile(filesim(handles.counter).folder,filesim(handles.counter).name);
i= imread(im);
imshow(i);
set(handles.nombre,'string',m1.nombre);
set(handles.edad,'string',m1.edad);
set(handles.sexo,'string',m1.sexo);
set(handles.direccion,'string',m1.direccion);
set(handles.telefono,'string',m1.telefono);
guidata(hObject, handles);
```

#### **Siguiente**

Este botón es usado para avanzar al siguiente registrado en la base de datos, en donde su código es el siguiente.

```
handles.counter = handles.counter + 1;
```

```
cla(handles.axes2, 'reset');
files = dir('C:\Users\user\Desktop\Basededatoss\**\*.mat');
fn=fullfile(files(handles.counter).folder,files(handles.counter).name);
m1=load(fn);
axes(handles.axes2);
filesim= dir('C:\Users\user\Desktop\Basededatoss\**\recortada1.bmp');
im=fullfile(filesim(handles.counter).folder,filesim(handles.counter).name);
i= inread(im);
imshow(i);
set(handles.nombre,'string',m1.nombre);
set(handles.edad,'string',m1.edad);
set(handles.sexo,'string',m1.sexo);
set(handles.direccion,'string',m1.direccion);
set(handles.telefono,'string',m1.telefono);
guidata(hObject, handles);
```

# Capítulo 4

## Pruebas y Resultados

*El presente capítulo muestra las pruebas que fueron realizadas con el prototipo, así como su funcionalidad con los entornos controlados y no controlados además del análisis de los datos y resultados que fueron obtenidos.*

### 4.1 Prueba con entorno controlado

En esta prueba se utilizaron 20 fotografías con compresión Haar wavelet del usuario 1 el cual se encuentra un entorno controlado, es decir no existen más rostros además del usuario mismo además de realizar diferentes poses tales como de frente, de perfil, con inclinación, utilizando lentes etc. Lo que da un panorama general de lo que podría pasar en la vida diaria. Vease imagen 4.1.



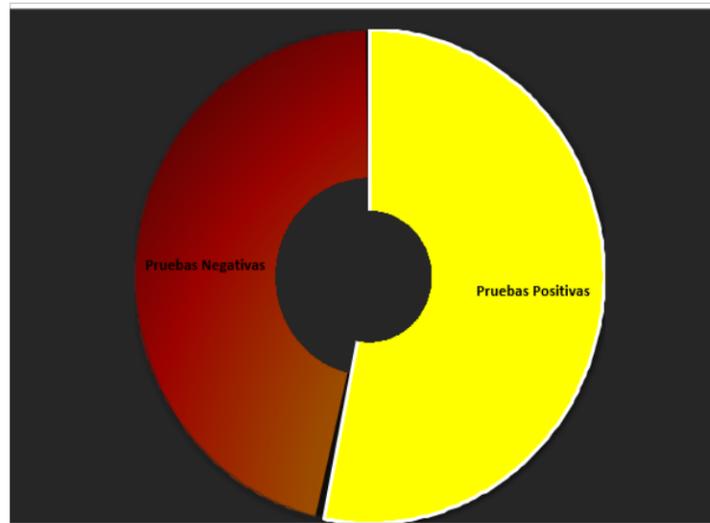
**Figura 4.1:** Imágenes capturadas del usuario 1 entorno controlado.

En la gráfica 4.1 se puede observar el porcentaje de pruebas negativas y positivas dadas por el prototipo dando un 55% de porcentaje positivo y un 45% negativo es importante mencionar que el algoritmo no reconoce el rostro cuando se encuentra de perfil

arrojándonos un 0% de porcentaje positivo cuando se encuentra en esta pose por lo que no es una opción que se debe utilizar en el prototipo.

**Tabla 4.1:** Resultados de usuario 1 con entorno controlado.

<b>TOTAL DE FOTOS</b>	<b>PRUEBAS POSITIVAS</b>	<b>PRUEBAS NEGATIVAS</b>
<b>20</b>	<b>11</b>	<b>9</b>



**Gráfica 4.1:** Resultados prueba 1

#### 4.1.1. Prueba con entorno controlado sujeto 2

En esta prueba se realizó el mismo procedimiento que con el usuario 1 ahora aplicándolos en el usuario 2 para poder determinar si a la persona a la que se le está aplicando el algoritmo es determinante para este ya que intervienen otros factores como geometría facial, ropa que utiliza etc. Véase figura 4.2.



**Figura 4.2:** Imágenes capturadas del usuario 2 en un entorno controlado

En la gráfica 4.2 se pudo observar que se obtuvo exactamente el mismo porcentaje de tanto pruebas negativas como positivas, comprobando que el usuario que utilice el prototipo no es relevante las características físicas que tenga o la ropa que utilice debido a que el algoritmo igualmente detecta el rostro.

**Tabla 4.2:** Resultados de usuario 2 con entorno controlado.

TOTAL DE FOTOS	PRUEBAS POSITIVAS	PRUEBAS NEGATIVAS
20	11	9



**Gráfica 4.2:** Resultados prueba con sujeto 2.

#### 4.2 Prueba con entorno no controlado

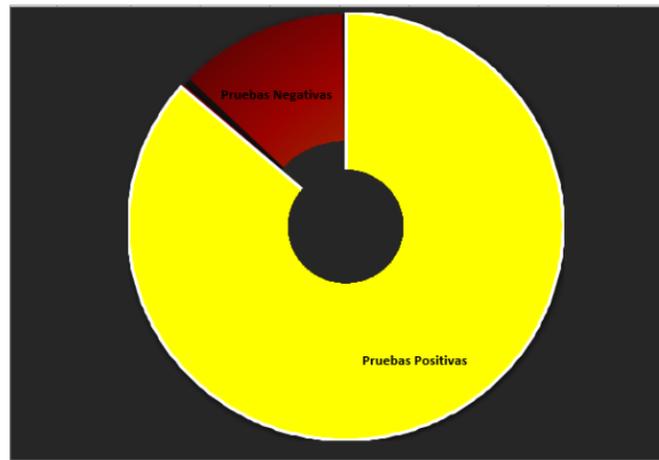
En esta prueba se tomaron 20 fotografías en un entorno en donde existiría más rostros existentes además del mismo usuario.



**Figura 4.3:** Imágenes capturadas con entorno no controlado.

**Tabla 4.3:** Resultados con entorno no controlado.

<b>TOTAL DE FOTOS</b>	<b>PRUEBAS POSITIVAS</b>	<b>PRUEBAS NEGATIVAS</b>
<b>20</b>	<b>17</b>	<b>3</b>



**Gráfica 4.3** Resultados de prueba entorno no controlado.

# Capítulo 5

## Conclusiones

### 5.1. Aportaciones

En este trabajo terminal se realizó una interfaz gráfica para el registro de los usuarios en una base de datos, realizando el registro mediante una webcam, para la captura de fotografías en el sistema, estas fotografías son comprimidas mediante el uso de transformada Wavelet logrando que la base de datos sea reducida en tamaño de almacenamiento significativamente.

Además, este sistema presenta dos alternativas de funcionamiento una en donde reforzamos la seguridad de un cajero automático mediante la activación de una alarma cuando existan más de dos usuarios queriendo hacer uso de este; la otra función se encarga de la detección y reconocimiento de usuarios empleando herramientas de programación en MATLAB, con una interfaz amigable tanto para el usuario como el administrador ya que el podrá guardar los datos y características de los usuarios en donde el considere sea más conveniente, teniendo la opción de tener la información en diferentes lugares y no concentrándola en un solo lugar permitiendo tener respaldos de seguridad en todo momento.

### 5.2. Trabajos futuros

Este trabajo tiene muchas opciones para ser empleado ya que podría ser utilizado en el sector salud, en donde para poder ser atendido es necesario contar con el carnet en la mano del beneficiario, al implementar este sistema en estos lugares, se reduciría en un tiempo considerable la burocracia existente además de que ya no sería necesario que el beneficiario tenga en su posesión el carnet físico ya que todos sus datos estarían ya registrados en una base nacional de salud.

Otra aplicación es utilizarlo cuando una alerta Amber sea activada.

Y la aplicación a la cual está enfocada este trabajo es el ser utilizada en la seguridad pública creando una base nacional de usuarios, la cual brinde información útil en investigaciones policíacas, haciendo uso de las cámaras del C4 y haciendo uso de este sistema optimizado se tendría la facultad de hacer reconocimiento facial en tiempo real para la detención de supuestos culpables, siendo una herramienta útil para los investigadores, por consiguiente México se modernizaría en este apartado tan crucial de la vida actual de la sociedad mexicana, el cual este tipo de sistemas ya ha sido demostrado su efectividad y repuesta en países de primer mundo como Estados Unidos, China y algunos países de la Unión Europea.

### 5.3 Conclusiones

La transformada wavelet de Haar es una herramienta matemática muy útil ya que ha demostrado ser superior respecto a la transformada de Fourier entregándonos más datos para ser utilizados que la transformada anteriormente dicha, ya que al utilizar la transformada wavelet de Haar de dos dimensiones la cual fue utilizada en imágenes, se comprobó que al utilizar solo el elemento B, la imagen no pierde características importantes y se logra realizar una compresión de la imagen en un tamaño significativo como se mostró en el capítulo 3.

En la parte de detección de rostros el algoritmo de Viola-jones a pesar de no ser perfecto mostro ventajas sobre otros algoritmos como por ejemplo el algoritmo de KLT(Kanade-Lucas-Tomasi), una de sus ventajas es el bajo costo computacional que requiere, brindando un mejor flujo de trabajo en el proceso de detección en presencia de una persona cuando esta se encuentre en frente de la cámara, además de que cuenta con otra ventaja la cual es no necesitar de sensores externos.

Se concluye también, que el algoritmo de Viola-Jones no es útil si el sujeto se encuentra de perfil, ya que en todas las pruebas realizadas en esta posición, resultaron negativas, por lo que el registro de usuarios tendrá que hacerse siempre viendo de frente a la cámara; Se observo que para tener un mejor funcionamiento es necesario hacer un procesamiento de imagen antes de extraer las características de HOG, como el tomar la imagen en una escala de grises y recortar la imagen para que solo se mostradas las características utilizable, en este caso el rostro.

El clasificador ECOC seleccionado para este trabajo con características HOG logro un buen rendimiento en la clasificación de sujetos, extrayendo las características extraídas de las fotografías tomadas para la base de datos.

Se concluye también que el umbral, el cual compara las mejores y peores puntuaciones para determinar el registro de un usuario, no es totalmente perfecto ya que en varias ocasiones termina arrojando datos erróneos, por lo que el registro de usuarios se tiene que hacer de una manera rigurosa y sistemática para evitar en la medida de los posibles los errores generados.

El diseño de la Interfaz Gráfica de Usuario (GUI) se realizó para que fuera totalmente amigable e intuitiva para cualquier administrador o usuario por lo tanto, la capacitación para el uso de este sistema sea prácticamente inexistente, sin olvidar que es un sistema seguro y confiable.

## Referencias bibliográficas

[Dip1] Rafael C. González, Digital Image Processing, Pearson, Segunda Edición, pp.498-450

[CIF00] John Miano ,Compressed Image File Formats, Addison Wesley Longman, pp. 24

[CIF01] John Miano ,Compressed Image File Formats, Addison Wesley Longman, pp. 24

[Dip2] Rafael C. González Digital ,Image Processing, Pearson, Segunda Edición, pp.427

[CIF06] John Miano ,Compressed Image File Formats, Addison Wesley Longman, pp. 37

[CIF07] John Miano ,Compressed Image File Formats, Addison Wesley Longman, pp. 9

[VF00] Patrick J. Van Fleet, Discrete Wavelet Transformations ,John Wiley & Sons, pp. 186-190, 2008

[CIF02] John Miano ,Compressed Image File Formats, Addison Wesley Longman, pp. 25

[CIF03] John Miano ,Compressed Image File Formats, Addison Wesley Longman, pp. 25

[VF01] Patrick J. Van Fleet, Discrete Wavelet Transformations ,John Wiley & Sons, pp. 158-166, 2008

[Pgl1]<https://www.pgj.cdmx.gob.mx/storage/app/media/Esta./2019/08-boletin-agosto-2019.pdf>

[BBC] <https://www.bbc.com/mundo/noticias-38610567>

[MAI1] Mohammad Ashraful islam, Md. Hasan; Comparision between viola-jones and klt algorithms and error correction of viola-jones algorithm , Mayo 17, [www.ijcea.com](http://www.ijcea.com)

[Pol1] <https://www.polibits.gelbukh.com>

[ESA01]<https://esacademic.com/dic.nsf/327025>

[GRA1]<https://www.aboutespanol.com/escala-de-grises-o-valor-180095>

[CRI01]<http://tecno logicamerani.edu.co/web/wp-content/uploads/2017/06/CRIMINALISTICA-AAV.pdf>

[CIF04] John Miano ,Compressed Image File Formats, Addison Wesley Longman, pp. 26

[CIF05] John Miano ,Compressed Image File Formats, Addison Wesley Longman, pp. 26