

**Universidad de las Ciencias Informáticas**

**Facultad # 1**

**Trabajo de Diploma para optar por el título de**

**Ingeniero en Ciencias Informáticas**

**Título:**

**Componente para la extracción de características en  
imágenes faciales**

**Autores:**

Anet González Jarrosay

Eider Barnet Díaz

**Tutor:**

Ing. Yainier Labrada Nueva

**Co-Tutor:**

Ing. Alejandro Lafourcade Despaigne

Junio, 2012

Año 54 de la Revolución

## DECLARACIÓN DE AUTORÍA

“Declaramos que somos los únicos autores del trabajo titulado: Componente para la extracción de características en imágenes faciales, y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo, con carácter exclusivo.”

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

Firma de los Autores:

\_\_\_\_\_  
Anet González Jarrosay

\_\_\_\_\_  
Eider Barnet Díaz

Firma de los tutores:

\_\_\_\_\_  
Ing. Yainier Labrada Nueva

\_\_\_\_\_  
Ing. Alejandro Lafourcade Despaigne

## **RESUMEN**

El reconocimiento facial basado en la apariencia ha mostrado un auge creciente en los últimos años debido a la disponibilidad de dispositivos de captura de imágenes, así como a la superioridad demostrada en la práctica de estos métodos con respecto a los métodos basados en rasgos faciales. En el presente trabajo se muestra el desarrollo de un componente que permite extraer características distintivas para completar el proceso de identificación biométrica basado en imágenes faciales en el departamento de Biometría, perteneciente al Centro de Identificación y Seguridad Digital de la Universidad de las Ciencias Informáticas. Para el desarrollo de dicho componente se utilizaron los principales algoritmos de extracción de características invariantes a la iluminación, según un análisis estadístico y el dominio de la frecuencia. Además, se proponen métodos para el reconocimiento facial de bajo costo computacional y robusto a cambios de iluminación, postura y expresión facial con el objetivo de ser capaz de identificar automáticamente a una persona mediante una imagen de entrada y comparándolas con imágenes ya existentes en una base de datos. Los métodos propuestos se encuentran divididos en tres etapas fundamentales: 1) extracción de características faciales por medio de la transformada discreta de wavelet (DWT), 2) aplicación del análisis de componentes principales (PCA) al vector extraído y 3) clasificación de patrones por medio de la distancia euclidiana a partir de los vectores característicos extraídos.

Palabras claves: Biometría, imagen facial, rasgo facial, reconocimiento facial.

---

**ÍNDICE**

<b>INTRODUCCIÓN .....</b>	<b>1</b>
<b>FUNDAMENTACIÓN TEÓRICA .....</b>	<b>2</b>
1.1 Introducción.....	2
1.2 Sistemas automatizados de reconocimiento facial .....	2
1.3 Procesos para el reconocimiento facial .....	3
1.4 Algoritmos para el reconocimiento automático facial .....	4
1.5 Extracción de características faciales.....	5
1.5.1 Métodos para la extracción de características faciales.....	5
1.5.2 Algoritmos para el reconocimiento automático facial basados en la apariencia.	6
1.5.2.1 Métodos de extracción de características según análisis estadístico.....	7
1.5.2.1.1 Análisis de Componentes Principales (PCA o <i>Eigenfaces</i> ).....	7
1.5.2.1.2 Análisis Lineal Discriminante (LDA o <i>Fisherfaces</i> ).....	9
1.5.2.2 Métodos de Extracción de Características según el Dominio de la Frecuencia.....	9
1.5.2.2.1 Transformada Discreta de Fourier.....	9
1.5.2.2.2 Fase del Dominio de Gabor .....	12
1.5.2.2.3 Transformada Discreta del Coseno (DCT) .....	13
1.5.2.2.4 Transformada Discreta de Wavelet (DWT) .....	15
1.5.2.2.4.1 Familia de funciones Wavelet .....	21
1.5.3 Análisis de los algoritmos estudiados .....	21
1.6 Metodologías, Lenguajes y Herramientas de Desarrollo .....	23
1.6.1 Metodologías de desarrollo de software .....	23
1.6.1.1 RUP .....	23
1.6.1.2 XP.....	24
1.6.1.3 Selección de la metodología.....	25
1.6.2 Lenguaje de Modelado .....	26
1.6.3 Herramientas de Modelado.....	27
1.6.3.1 Herramienta CASE .....	27

---

1.6.3.1.1	Visual Paradigm para UML .....	27
1.6.4	Lenguaje de Programación .....	27
1.6.4.1	Lenguaje Matlab .....	28
1.6.5	Entorno Integrado de Desarrollo .....	28
1.7	Conclusiones Parciales .....	30
<b>ANÁLISIS Y DISEÑO .....</b>	<b>31</b>	
2.1	Introducción .....	31
2.2	Exploración.....	31
2.2.1	Descripción del sistema .....	32
2.2.2	Arquitectura de la solución.....	32
2.2.3	Usuarios del Sistema .....	33
2.2.4	Historias de Usuario.....	34
2.2.5	Especificación de los requisitos de software.....	36
2.2.6.1	Requisitos no funcionales.....	37
2.2.6.1.1	Software .....	37
2.2.6.1.2	Requisitos de hardware mínimo .....	37
2.2.6.1.3	Rendimiento .....	37
2.2.6.1.4	Requisitos de seguridad.....	38
2.3	Planificación .....	38
2.3.1	Estimación de esfuerzo por Historias de Usuario .....	38
2.3.2	Plan de Iteraciones .....	39
2.3.2.1	Plan de duración de las iteraciones.....	39
2.3.3	Plan de entregas.....	40
2.4	Conclusiones Parciales .....	40
<b>IMPLEMENTACIÓN Y PRUEBA.....</b>	<b>41</b>	
3.1	Introducción.....	41
3.2	Fase de Diseño. ....	41
3.2.1	Funciones implementadas para el desarrollo del componente .....	43
3.2.2	Principales funciones utilizadas del Matlab.....	44
3.2.3	Estándar de programación.....	45

3.2.4	Tareas de la Ingeniería .....	47
3.2.5	Requisitos para la integración del componente .....	49
3.3	Pruebas.....	51
3.3.1	Pruebas de aceptación. ....	51
3.4	Conclusiones Parciales .....	58
<b>CONCLUSIONES GENERALES .....</b>		<b>59</b>
<b>RECOMENDACIONES .....</b>		<b>60</b>
<b>REFERENCIA BIBLIOGRÁFICA.....</b>		<b>61</b>
<b>GLOSARIO DE TÉRMINOS.....</b>		<b>64</b>
<b>ANEXOS.....</b>		<b>66</b>

## **ÍNDICE DE FIGURAS**

FIGURA 1. EJEMPLOS DE RASGOS BIOMÉTRICOS.....	1
FIGURA 2. PROCESOS PARA EL RECONOCIMIENTO FACIAL. ....	4
FIGURA 3. ALGORITMOS DE EXTRACCIÓN DE CARACTERÍSTICAS INVARIANTES A LA ILUMINACIÓN.....	6
FIGURA 4. ESQUEMA DE LA TRANSFORMADA DE FOURIER (17). ....	10
FIGURA 5. ESQUEMA DE LA TRANSFORMADA DE FOURIER POR INTERVALOS (STFT) (17).....	12
FIGURA 6. REPRESENTACIÓN DE LA EXTRACCIÓN DE RASGOS DCT UTILIZANDO RECORRIDO EN ZIGZAG (19). .....	13
FIGURA 7. INFLUENCIA DE LAS VARIACIONES DE ILUMINACIÓN EN LOS COEFICIENTES DCT DE BAJAS FRECUENCIAS DE DOS IMÁGENES DE UN MISMO ROSTRO (19). ....	15
FIGURA 8. ESQUEMA DE FILTRADO PARA CALCULAR LA DWT UNIDIMENSIONAL (14). ....	17
FIGURA 9. ÁRBOL DE DESCOMPOSICIÓN POR WAVELETS EN TRES NIVELES (14). ....	18
FIGURA 10. ESQUEMA DE FILTRADO PARA CALCULAR LA DWT DE UNA IMAGEN (14). ....	18
FIGURA 11. DESCOMPOSICIÓN DE LA IMAGEN FACIAL EN MATRICES DE APROXIMACIÓN Y DETALLES (14). ..	19
FIGURA 12. ESQUEMA DE EXTRACCIÓN DE CARACTERÍSTICAS MEDIANTE LA DWT (14).....	20
FIGURA 13. FASES DEL PROCESOS UNIFICADO DE RUP (24). ....	24
FIGURA 14. REPRESENTACIÓN DE LA METODOLOGÍA XP (26). ....	25
FIGURA 15 ARQUITECTURA TUBERÍAS Y FILTROS.....	33
FIGURA 16. DIAGRAMA DE COMPONENTE. ....	43
FIGURA 17. FAMILIAS DE WAVELET EN EL NIVEL 4 DE DESCOMPOSICIÓN. ....	57
FIGURA 18. FAMILIAS DE WAVELET EN NIVELES DEL 1 AL 4.....	58
FIGURA 19. DECIMACIÓN EN TIEMPO. ....	66

**ÍNDICE DE TABLAS**

TABLA 1. ACTOR DEL SISTEMA. ....	33
TABLA 2. TRANSFORMAR IMÁGENES A ESCALA DE GRISES. ....	34
TABLA 3. APLICAR TRANSFORMADA DISCRETA DE WAVELET. ....	34
TABLA 4. APLICAR ANÁLISIS DE COMPONENTES PRINCIPALES (PCA). ....	35
TABLA 5. EXPORTAR VECTOR CARACTERÍSTICO. ....	35
TABLA 6. CALCULAR DISTANCIA EUCLIDIANA. ....	36
TABLA 7. ESTIMACIÓN DE ESFUERZO POR HISTORIAS DE USUARIO. ....	38
TABLA 8. PLAN DE DURACIÓN DE LAS ITERACIONES. ....	39
TABLA 9. PLAN DE ENTREGA. ....	40
TABLA 10. HU1_TRANSFORMAR IMÁGENES A ESCALA DE GRISES Y MOSTRAR LA IMAGEN DE PRUEBA TRANSFORMADA. ....	47
TABLA 11. HU2_DESARROLLAR EL ALGORITMO DE LA TRANSFORMADA DISCRETA DE WAVELET. ....	48
TABLA 12. HU3_DESARROLLAR EL ALGORITMO ANÁLISIS DE COMPONENTES PRINCIPALES (PCA).....	48
TABLA 13. HU4_EXPORTAR VECTOR CARACTERÍSTICO. ....	48
TABLA 14. HU5_CALCULAR DISTANCIA EUCLIDIANA. ....	49
TABLA 15. CP1_TRANSFORMAR LAS IMÁGENES A ESCALA DE GRISES. ....	52
TABLA 16. CP2_APLICAR TRANSFORMADA DISCRETA DE WAVELET. ....	52
TABLA 17. CP3_APLICAR EL ANÁLISIS DE PRINCIPALES COMPONENTES (PCA). ....	53
TABLA 18. CP4_EXPORTAR VECTOR CARACTERÍSTICO. ....	54
TABLA 19. CP5_MOSTRAR RESULTADOS DEL RECONOCIMIENTO. ....	54
TABLA 20. CP6_PROBAR COSTO COMPUTACIONAL DE VECTORES WAVELET Y PCA. ....	55
TABLA 21. CP7_FAMILIA WAVELET. ....	56
TABLA 22. CP8_NIVEL DE DESCOMPOSICIÓN DE WAVELET .....	57



## INTRODUCCIÓN

En la actualidad, debido a los avances tecnológicos y la creciente demanda de sistemas de seguridad y video vigilancia, las aplicaciones biométricas han tomado un gran auge. La biometría es la disciplina que permite identificar a los individuos basándose en sus características físicas y en su comportamiento. Entre las principales características físicas se pueden mencionar las huellas dactilares, el iris, la retina, la geometría de la palma de la mano y los rasgos faciales, entre otras. Por otra parte, dentro de las características del comportamiento más comunes se encuentran la firma manuscrita, la forma de caminar, la voz y la pulsación en teclado.



Figura 1. Ejemplos de rasgos biométricos.

El reconocimiento automático de rasgos faciales es utilizado en numerosas aplicaciones prácticas, como por ejemplo sistemas de vigilancia, control de inmigración, control de acceso, autenticación de usuarios para dispositivos electrónicos como teléfonos celulares, cámaras, agendas electrónicas, etc. Es uno de los métodos biométricos más usados debido entre otros factores a que es una técnica no invasiva, natural y fácil de usar (1).

Cuba no ha estado ajena al desarrollo de la biometría, la cual se manifiesta en un sinnúmero de aplicaciones para la investigación criminal, la protección y la seguridad. Demostrando ser un novedoso método de identificación humana por excelencia, de ahí el especial interés de propiciar el intercambio entre profesionales cubanos y extranjeros con reconocida pericia en el desarrollo de aplicaciones biométricas. Actualmente existe en Cuba el Centro de Aplicaciones de Tecnologías de Avanzadas (CENATAV), donde se hacen estudios de los patrones biométricos para el reconocimiento de personas según sus características faciales. CENATAV es un centro de investigaciones teóricas y aplicadas que tiene como misión fundamental asimilar, desarrollar e introducir en la práctica social los aspectos más novedosos de la teoría y la práctica del reconocimiento de patrones (RP) en su concepción más general, y de la minería de datos (MD), que permitan responder a las necesidades del progreso científico-técnico y socio-económico, así como incrementar el patrimonio científico nacional (2).

En la Universidad de las Ciencias Informáticas (UCI) se encuentra el Centro de Identificación y Seguridad Digital (CISED), el cual cuenta con el departamento de Biometría, que se encarga de estudiar los patrones para el reconocimiento único de personas basados en rasgos físicos o de conducta, como: huellas dactilares, firma manuscrita y reconocimiento facial. En dicho departamento se pretende implementar las funcionalidades concernientes a los procesos de reconocimiento facial. En el mismo se han desarrollado componentes para los subprocesos de captura, procesamiento, normalización y validación de la calidad de la imagen, sin embargo, el proceso de reconocimiento facial no está totalmente completo debido a que no se realizan subprocesos como la extracción de características faciales de una imagen.

Planteada la situación problemática puede enunciarse el **problema de la investigación** a partir de la siguiente interrogante ¿Cómo extraer características en imágenes faciales para el proceso de reconocimiento de rostros en el departamento de Biometría?

Se ha determinado como **objeto de estudio** del presente trabajo de diploma es el proceso de extracción de características en imágenes.

Para dar solución al problema de la investigación se plantea como **objetivo general** desarrollar un componente de extracción de características distintivas para su uso como elemento de comparación en procesos de identificación biométrica basados en imágenes faciales.

El desarrollo del componente tendrá como **campo de acción** la extracción de características en imágenes faciales utilizando técnicas basadas en la apariencia facial.

Los **objetivos específicos** que se persiguen son:

- Elaborar el marco teórico de la investigación. (Anet González Jarrosay)
- Desarrollar el componente para realizar el proceso de extracción de características en imágenes faciales. (Eider Barnet Díaz y Anet González Jarrosay)
- Validar la solución del componente de extracción de características. (Eider Barnet Díaz)

Para dar cumplimiento a los objetivos planteados se planificaron **tareas investigativas**:

- Definición de los diferentes conceptos de extracción de características en imágenes faciales. (Anet González Jarrosay)
- Análisis y selección de los distintos algoritmos existentes para la extracción de características en imágenes faciales. (Eider Barnet Díaz)
- Análisis y selección de las tecnologías y herramientas que permitan la extracción de características en imágenes faciales y el desarrollo de un componente para realizar dicho proceso. (Anet González Jarrosay)
- Implementación del algoritmo seleccionado para la fase de extracción, incorporando dicha funcionalidad en un componente que muestre su desempeño. (Eider Barnet Díaz y Anet González Jarrosay)
- Realización de las pruebas al componente de extracción de características en imágenes faciales, para verificar su correcto funcionamiento. (Eider Barnet Díaz)

Se utilizaron **métodos científicos** para el desarrollo de la investigación y lograr caracterizar a fondo el objeto de estudio, así como garantizar el conocimiento del estado del arte, su evolución y relación con otros fenómenos.

Dentro de los métodos científicos utilizados se encuentra el **histórico-lógico** el cual posibilita la comprensión lógica del objeto de estudio haciendo un análisis riguroso de sus antecedentes y el proceso evolutivo por el cual han transitado todas las tecnologías relacionadas con el reconocimiento facial.

Se utiliza el método **analítico-sintético** que permite la consulta de diversas fuentes bibliográficas y la extracción de los elementos más importantes que se relacionan con el objeto de estudio.

La **justificación de la investigación** del trabajo se basa en el desarrollo de un componente que permita realizar la extracción de características en imágenes faciales. Además, forma parte de los esfuerzos del CISED para crear sistemas propios de identificación, representando un paso importante para encontrar soluciones alternativas y de buena calidad a los costosos sistemas de identificación existentes en el mercado.

El contenido está estructurado en tres capítulos los cuales se agrupan de la manera siguiente:

**Capítulo 1: Fundamentación teórica** incluye un estudio del estado del arte del tema tratado; se abordan elementos teóricos de la investigación tales como biometría, proceso de reconocimiento facial, metodología, lenguajes y herramientas de desarrollo que serán utilizadas para implementar la solución.

**Capítulo 2: Análisis y diseño** se presentan las fases de Exploración y Planificación definidas por la metodología Programación Extrema (XP) por sus siglas en inglés, para dar solución al problema científico. Se selecciona la arquitectura para desarrollar la solución. Se identifican las historias de usuarios y los requisitos no funcionales, se realiza el plan de iteraciones y plan de entregas.

**Capítulo 3: Implementación y prueba** se dan cumplimiento a los planes trazados a través de las fases: diseño y prueba, se codifica la solución diseñada y finalmente se realizan las pruebas de aceptación.

# Capítulo 7

## FUNDAMENTACIÓN TEÓRICA

### 1.1 Introducción

En este capítulo se definen un conjunto de ideas que enmarcan al lector en el contexto en el cual se desarrolla la investigación. Se pretende lograr un basamento teórico sólido a partir de los conceptos y definiciones adquiridos en el estudio del estado del arte y las tendencias actuales en el desarrollo de sistemas automatizados de reconocimiento facial. Por otra parte, se realiza un análisis comparativo de las principales metodologías existentes, además de hacer un estudio detallado de las tecnologías y herramientas que se utilizan a lo largo de la presente investigación y su desarrollo. Se realiza una breve descripción del proceso de extracción de características en imágenes faciales y se describen los algoritmos para el reconocimiento automático facial, así como los métodos para la extracción de características faciales en imágenes.

### 1.2 Sistemas automatizados de reconocimiento facial

Los Sistemas de Reconocimiento Facial son un concepto desarrollado a partir de los años 60, cuando un grupo de científicos norteamericanos desarrollaron el primer sistema semiautomático para reconocimiento facial que requería del usuario para localizar características faciales como los ojos, las orejas, la nariz y la boca en las fotografías. A partir de los puntos seleccionados, se calculaban las distancias entre ellos y eran comparadas con datos de referencia almacenados previamente (3). En la década del 70, Goldstein<sup>1</sup> y su grupo agregaron 21 marcadores subjetivos específicos, entre los que se encontraban el color del cabello y el grosor de los labios, con el objetivo de contar con mayor información para la comparación (4). Estas primeras técnicas, basadas en la geometría de las características faciales, no tuvieron mucho éxito en la práctica, por lo engorroso que resulta la anotación manual del gran número de imágenes faciales que normalmente se procesan.

A principios de los años 90, los científicos Brunelli y Poggio presentaron un sistema de reconocimiento facial, que extraía automáticamente 35 características faciales correspondientes a

---

<sup>1</sup> Científico norteamericano.

puntos de los ojos, la nariz, la boca y la barbilla. Con esto se formaba un vector de 35 dimensiones para representar el rostro, que posteriormente se utilizaba para determinar la correspondencia con otros rostros por medio de un clasificador Bayesiano (5). Los resultados reportados por este sistema fueron de un 90 % de clasificación correcta para una base de datos de 47 individuos, siendo muy bajo el costo computacional y de almacenamiento del sistema. A pesar de los buenos resultados experimentales obtenidos, lo fundamental en este sistema era la localización automática correcta de las características distintivas faciales, lo cual, incluso en la actualidad es una tarea complicada, principalmente cuando hay variaciones de pose, iluminación u oclusión en las imágenes.

Con los métodos más modernos las tasas de error en la localización de los rasgos faciales cuando existen distintas variaciones son aún altas. Por otra parte, al representar el rostro solamente por sus características faciales, se pierde información importante sobre la textura y los tonos presentes en la imagen (6) (7) (8).

Lo expuesto anteriormente dio origen a las llamadas técnicas basadas en la apariencia, que significaron un gran avance para la eficacia y eficiencia de los sistemas de reconocimiento facial que operan directamente con las intensidades de los píxeles u otras representaciones basadas en la imagen como las que son utilizadas como rasgos para la clasificación, empleándose para ello herramientas inteligentes de diversas áreas del conocimiento como el reconocimiento de patrones y las redes neuronales.

### **1.3 Procesos para el reconocimiento facial**

Los Sistemas de Reconocimiento Facial operan de dos formas diferentes: verificación (autenticación) e identificación. En un sistema de verificación, se realiza una solicitud de identidad al usuario por medio de un identificador artificial (contraseña, tarjeta, etc.), luego la identidad es verificada capturando la muestra biométrica del usuario y comparándola con la muestra almacenada que corresponde a dicho identificador. Por su parte, los sistemas de identificación capturan la muestra biométrica del usuario y determinan, mediante una búsqueda en una base de datos de una muestra biométrica aceptablemente aproximada, la identidad del usuario.

El reconocimiento facial es una tarea compuesta por varios procesos, por lo general consiste en la detección, procesamiento, extracción de características y clasificación. La detección facial segmenta la región de la imagen en la que se encuentra localizado el rostro. En el procesamiento se realizan la validación y normalización de la imagen facial para mejorar el rendimiento del componente. En la

extracción de características se obtiene una representación de baja dimensionalidad de la imagen facial mediante el mapeo del espacio original de la imagen a un subespacio característico más simple y menos sensible a variaciones no relacionadas a la identidad. La clasificación es la última etapa de un sistema típico de reconocimiento facial, en esta etapa el subespacio característico es particionado y los vectores característicos son clasificados de acuerdo a su identidad.

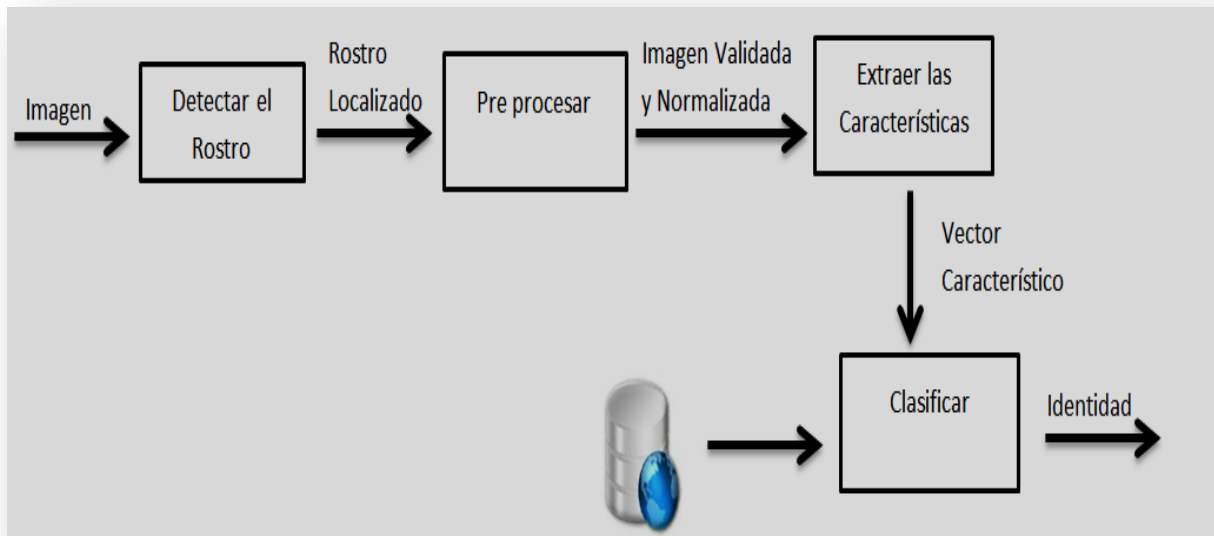


Figura 2. Procesos para el Reconocimiento Facial.

#### 1.4 Algoritmos para el reconocimiento automático facial

Los algoritmos de reconocimiento automático facial constan de un conjunto de pasos que se ejecutan a partir de que se tiene una imagen digital en la que se encuentra el rostro de la persona que se quiere reconocer. El primero de estos pasos consiste en la detección facial dentro de la imagen. El segundo paso es la alineación o normalización, que incluye transformaciones geométricas y correcciones de la iluminación, con el objetivo de estandarizar las diferentes imágenes faciales. Una vez que las imágenes están normalizadas se procede a extraer los rasgos faciales que son usados para el reconocimiento, representados normalmente como un vector de características. Finalmente, tiene lugar la clasificación, que no es más que comparar la información extraída en el paso anterior, contra uno o más modelos de rostros, para finalmente obtener una respuesta sobre la imagen facial que se está analizando (9).

Los algoritmos para el reconocimiento automático facial se clasifican en: basados en características geométricas y en la apariencia. Los algoritmos basados en características geométricas requieren la localización exacta de los rasgos faciales, lo cual aún no logra hacerse de manera precisa con métodos automáticos (6) (7) (8). Los algoritmos de reconocimiento facial basados en la apariencia han sido los predominantes en los últimos años. Estos pueden utilizar la información del rostro de manera global o de manera local. Los que se basan en representaciones globales, identifican un rostro utilizando para ello un vector que representa la imagen completa, mientras que los basados en métodos locales usan vectores de rasgos que representan localidades de la imagen de rostro (10) (11).

## **1.5 Extracción de características faciales**

La utilización de las características faciales se ha convertido en un reto en la actualidad, especialmente en aplicaciones donde existen ambientes no controlados donde inciden negativamente factores sobre las potencialidades biométricas de los rostros humanos, como son: la pose, la iluminación, la expresión, los accesorios utilizados, entre otros. Debido a esto se ha analizado la influencia de cada uno de ellos en el rendimiento de los sistemas de reconocimiento facial, determinándose que las variaciones de pose e iluminación, son las que en mayor medida los afectan.

### **1.5.1 Métodos para la extracción de características faciales**

Existen varios métodos para la extracción de características y se dividen en varios subconjuntos en dependencia de las herramientas utilizadas para el análisis de la imagen. En esta investigación se utilizaron dos subconjuntos: según un análisis estadístico y el dominio de la frecuencia. En el primero se encuentran agrupados los algoritmos que utilizan técnicas estadísticas para obtener una representación de la imagen que no dependa de las condiciones de iluminación. El segundo está compuesto por los algoritmos que transforman la imagen al dominio de la frecuencia utilizando diferentes filtros o transformaciones; en este dominio se define la correspondencia entre las imágenes faciales.



### 1.5.2 Algoritmos para el reconocimiento automático facial basados en la apariencia.

Existen varios algoritmos pertenecientes a los métodos que realizan análisis estadísticos. Para el desarrollo del componente de extracción de características en imágenes faciales se estudiaron los algoritmos Análisis de Componentes Principales (PCA o *Eigenfaces*) y Análisis de Discriminante Lineal (LDA), debido a que estos son los algoritmos más usados en el mundo para este tipo de métodos. En el caso del método del dominio de la frecuencia, se analizaron los algoritmos Transformada Discreta de Fourier (*Corefaces*), Fase del Dominio de Gabor, Transformada Discreta del Coseno (DCT) y Transformada Discreta de Wavelet (DWT).



Figura 3. Algoritmos de extracción de características invariantes a la iluminación.

### 1.5.2.1 Métodos de extracción de características según análisis estadístico

Los métodos que se ubican en este conjunto se basan en extraer los rasgos representativos del rostro con herramientas estadísticas a partir de la información de la intensidad de los píxeles de la imagen. Estas herramientas se basan en propiedades que permiten establecer relaciones entre los píxeles de la imagen, y que por otro lado, pueden utilizarse para reducir la dimensión de la representación de la imagen facial (12).

#### 1.5.2.1.1 Análisis de Componentes Principales (PCA o *Eigenfaces*)

El objetivo del algoritmo Análisis de Componentes Principales (PCA) por sus siglas en inglés, es extraer la información relevante en una imagen facial y codificarla lo más eficientemente posible. Esta es una técnica de reducción de la dimensión lineal no supervisada que hace un análisis global de los datos hallando la varianza total del sistema (13). Luego encuentra una representación en un espacio de menor dimensión que conserva la mayor cantidad de varianza posible. El procedimiento es sencillo, primero se representan las imágenes faciales como vectores. Sea  $X = [x_1 \dots x_N]$ , la matriz donde cada columna es una representación vectorial de la imagen facial conteniendo los píxeles de la imagen. Luego se halla el rostro promedio del conjunto que se define como:

$$\Psi = \frac{1}{N} \sum_{i=1}^N X_i \quad (14)$$

Cada rostro difiere del promedio por el vector  $\phi_i = X_i - \Psi$  (14). Este conjunto de vectores extremadamente grandes es sometido al análisis de componentes principales, el cual busca el conjunto de vectores ortogonales  $\mu_n$  que mejor describe la distribución de los datos.

Luego se define la matriz de covarianza C como:

$$C = \frac{1}{M} \sum_{n=1}^M \phi_n \phi_n^T = AA^T \quad (14)$$

Sin embargo, la matriz C es de  $N^2 \times N^2$ , y determinar los  $N^2$  eigenvectores y eigenvalores es una tarea intratable para las dimensiones típicas de las imágenes. Si el número de puntos de datos en el espacio de imagen es menor que la dimensión del espacio ( $M < N^2$ ), habrá únicamente  $M - 1$  en lugar de  $N^2$  eigenvectores significativos (el resto de los eigenvectores tendrán eigenvalores asociados iguales a cero). Afortunadamente es posible resolver este problema para los eigenvectores de dimensión  $N^2$ , en este caso, primero se resuelve para los eigenvectores de una matriz de  $M \times M$  y consecuentemente se toman combinaciones lineales apropiadas de las imágenes faciales  $\phi_i$ .

Considere los eigenvectores  $v_i$  de  $A^T A$  tal que

$$A^T A v_i = \mu_i v_i \quad (14)$$

Multiplicando ambos lados por  $A$  tenemos

$$A A^T A v_i = \mu_i A v_i \quad (14)$$

De donde se puede ver que  $v_i$  son los eigenvectores de  $C = A A^T$ .

Siguiendo con este análisis, se construye la matriz  $L = A^T A$  de  $M \times M$ , donde se encuentran los  $M$  eigenvectores de  $L$ .

Estos vectores determinan combinaciones lineales de las imágenes faciales del conjunto de entrenamiento para formar las *eigenfaces*  $u_l$ .

$$u_l = \sum_{k=1}^M v_{lk} \Phi_k, \quad l = 1, \dots, M \quad (14)$$

Con este análisis los cálculos son reducidos enormemente del orden del número de píxeles en las imágenes ( $N^2$ ) al orden del número de imágenes en el conjunto de entrenamiento ( $M$ ).

Una vez que las *eigenfaces* han sido calculadas, la imagen es proyectada en el espacio facial mediante una simple operación definida como

$$\omega_k = u_k^T (X - \Psi) \quad (14)$$

Para  $k = 1, \dots, M$ . Los pesos forman un vector

$$\Omega^T = [\omega_1, \omega_2, \dots, \omega_M] \quad (14)$$

Que describe la contribución de cada *eigenface* en la imagen facial de entrada.

Por último, el método más simple para determinar cuál de los rostros almacenados proporciona la mejor descripción de la imagen facial de entrada es encontrando el rostro que minimice la distancia Euclidiana.

$$\epsilon_k^2 = \|\Omega - \Omega_k\|^2 \quad (14)$$

Donde  $\Omega_k$  es un vector que describe la  $k$ -ésima clase de rostro.

### 1.5.2.1.2 Análisis Lineal Discriminante (LDA o Fisherfaces)

El algoritmo Análisis lineal discriminante (*Linear Discriminant Analysis*) LDA por sus siglas en inglés, es una técnica supervisada de reducción lineal de la dimensión que pretende hallar un subespacio discriminante. El objetivo de sus creadores es que los elementos pertenecientes a clases diferentes, en este caso rostros de personas diferentes, se mantengan alejados en el nuevo espacio, mientras que los elementos pertenecientes a la misma clase permanezcan cercanos. Esencialmente se halla la matriz de dispersión intraclase<sup>2</sup> e interclase<sup>3</sup> suponiendo que bajo condiciones ideales las clases son linealmente separables, lo que significa ejecutar la clasificación en el espacio de rasgos de menor dimensión con fronteras de decisión lineales (12).

Entre las ventajas principales se puede mencionar el bajo costo computacional, ya que una vez que se tienen las direcciones calculadas para cada clase, solo es necesaria una imagen para proyectar y emplear dicha proyección como modelo de la clase. Sin embargo, tiene como desventajas se encuentra el alto costo del entrenamiento, y que requiere condiciones difíciles de hallar en la práctica, como un conjunto de imágenes por cada clase con diferentes condiciones de iluminación (12).

### 1.5.2.2 Métodos de Extracción de Características según el Dominio de la Frecuencia

Los métodos que pertenecen a este subconjunto se caracterizan por aplicar transformaciones a la imagen y obtener una representación de la misma en el dominio de la frecuencia. Se verán cuatro técnicas que transforman la imagen al dominio de la frecuencia, dado que es más fácil separar los efectos de la iluminación en este dominio, estando mayormente presentes las variaciones de iluminación en las bajas frecuencias, siendo poco predominantes en las altas frecuencias (15).

#### 1.5.2.2.1 Transformada Discreta de Fourier

La función de la transformada de Fourier se desarrolla en el contexto del dominio de la frecuencia, para la eliminación de bajas y altas frecuencias, lo que se traduce como suavizar la imagen y la detección de bordes.

Como herramienta matemática es usada en una gran variedad de aplicaciones para el procesamiento digital de imágenes y desarrollo de algoritmos para la implementación. Se encuentra

---

<sup>2</sup> Imágenes pertenecientes a diferentes personas bajo una misma condición de iluminación, pose y expresión facial.

<sup>3</sup> Imágenes pertenecientes a una misma persona personas bajo diferentes condiciones de iluminación, pose y expresión facial.

implementada bajo la forma de dispositivos electrónicos de reconocimiento de voz e imagen; puede ser aplicada a varios campos como análisis espectral, ecuaciones diferenciales, resolución de problemas estacionarios, elásticos y dinámicos.

Una de sus aplicaciones principales es en el diseño de algoritmos de filtrado para el procesamiento, compresión, restauración y mejora de imágenes, acompañado de la transformada inversa que realza las frecuencias espaciales altas, de modo que cuando es aplicado a imágenes que contienen formas geométricas sencillas permite realizar un proceso de segmentación, cuando es aplicada a imágenes con bordes poco diferenciados produce un realce de los mismos y cuando es aplicada a imágenes con ruido produce una reducción.

La Transformada de Fourier es una de las más importantes herramientas que ha sido usada no solamente para entender la naturaleza de una imagen y su formación sino también para procesarla. Una imagen es una señal bidimensional y puede ser vista como una superficie en un espacio bidimensional (16).

Esta transformación permite descomponer una señal en sus componentes sinusoidales de diferentes frecuencias, en otras palabras, puede pensarse que es una técnica matemática para transformar el punto de vista de una señal desde la base de tiempo a la base de la frecuencia, tal como se representa esquemáticamente en la Figura 4 (17).



Figura 4. Esquema de la Transformada de Fourier (17).

Para un mejor funcionamiento de Fourier la señal que se está analizando no debe cambiar demasiado en el tiempo, es decir, debe ser una señal estacionaria<sup>4</sup>. Sin embargo, un importante número de señales de interés presentan características no estacionarias o transitorias, tales como una tendencia, cambios abruptos, comienzos o finales de eventos o que presentan cambios bruscos en intervalos muy pequeños. A menudo, estas características no estacionarias resultan ser las secciones más interesantes de las señales, y la Transformada de Fourier no está preparada para detectarlas y analizarlas.

Una imagen se puede filtrar en el dominio de la frecuencia o en el dominio del espacio. Los filtros en el dominio de la frecuencia se usan, principalmente, para eliminar altas o bajas frecuencias de la imagen, lo que se traduce en suavizar la imagen, o bien, realzar o detectar bordes.

Básicamente, hay tres tipos diferentes de filtros de frecuencia:

- Filtros de paso bajo (*lowpass filter*). Deja inalterables las bajas frecuencias y atenúa o elimina las altas frecuencias. El resultado en el dominio del espacio consiste en un suavizado: eliminar pequeños detalles y ruidos de la imagen. Este filtro suprime las altas frecuencias mayores que un cierto valor  $D_0$ , que se denomina frecuencia de corte, y deja las demás tal como están. (18)
- Filtros de paso alto (*highpass filter*). Deja inalterables las altas frecuencias y atenúa o elimina las bajas frecuencias. El resultado en el dominio del espacio consiste en un realzado de los cambios bruscos de niveles de grises. De ahí que se use para detectar bordes. Las áreas de niveles de gris constantes o con poca variación se corresponden con las bajas frecuencias, que se suprimen. Este filtro suprime las frecuencias menores o iguales que un cierto valor  $D_0$ , que se denomina frecuencia de corte (18).
- Filtros de banda (*bandpass filter*). Un filtro de banda atenúa las altas y bajas frecuencias, pero mantiene intactas las frecuencias que se encuentren en una banda determinada. Se puede obtener un filtro de banda multiplicando uno de paso bajo por uno de paso alto, en el que la frecuencia de corte del de paso bajo sea superior a la del de paso alto. El opuesto al filtro de paso de banda sería de “rechazo de banda” (*bandreject* o *band-stop*), en el que se atenúan las frecuencias de la banda, pero se mantienen la frecuencias fuera de ella (18).

---

<sup>4</sup> En el caso de las señales estacionarias, no interesa saber cuándo ocurre una determinada frecuencia, ya que estas ocurren siempre.

### 1.5.2.2 Fase del Dominio de Gabor

En un esfuerzo por corregir la deficiencia presentada por Fourier, en 1946 Denis Gabor adaptó la Transformada de Fourier para poder analizar una pequeña sección de la señal en un determinado tiempo (mediante una especie de ventana). Esta adaptación es la que se conoce como la Transformada de Fourier en corto tiempo (*Short time Fourier transform*) STFT por sus siglas en inglés, la cual lleva una señal del plano del tiempo al plano bidimensional de tiempo y frecuencia, tal como se presenta esquemáticamente en la Figura 5 (17).



Figura 5. Esquema de la Transformada de Fourier por intervalos (STFT) (17).

Es importante mencionar que la STFT representa una especie de compromiso entre el dominio del tiempo y el de la frecuencia de una señal, ya que provee algo de información acerca de cuándo y a qué frecuencia de una señal ocurre un determinado evento. Sin embargo, solamente se puede obtener dicha información con una precisión limitada, la cual está acotada por el tamaño de la ventana. El mismo concepto puede ser extendido al espacio bidimensional de la imagen donde los componentes de la frecuencia localizada pueden ser determinados por la ventana de la transformada.

Mientras que el compromiso entre la información del tiempo y la frecuencia puede resultar útil, el inconveniente surge dado que una vez que se escoge un determinado tamaño para la ventana de tiempo, dicha ventana es la misma para todas las frecuencias. Muchas señales requieren un acercamiento más flexible, de modo tal que sea posible variar el tamaño de la ventana para determinar con mayor precisión el tiempo o la frecuencia (17).

Resulta de utilidad en los problemas de análisis y segmentación de la imagen, ya que extrae localmente la información espectral bidimensional concerniente a la textura, cosa que no es posible con la Transformada de Fourier.

### 1.5.2.2.3 Transformada Discreta del Coseno (DCT)

La Transformada Discreta del Coseno (DCT) ha sido usada como un paso en la extracción de características en diferentes estudios de reconocimiento facial, tanto en algoritmos basados en la apariencia global como en otros basados en la apariencia local. Principalmente han sido clasificadas cuatro transformaciones ligeramente diferentes: DCTI, DCTII, DCT III y DCT IV. De ellas, La DCT II ha sido la más utilizada en el procesamiento de imágenes.

En “*Local appearance based face recognition using discrete cosine transform*” se propone un método de reconocimiento facial basado en la DCT, en el cual se divide la imagen en bloques de píxeles. Seguidamente, la DCT de cada bloque de la imagen es calculada y los coeficientes obtenidos se reducen eliminando los componentes en las bandas de frecuencia más altas y posteriormente son normalizados, estos valores se concatenan para formar el vector característico.

Este vector contiene los coeficientes DCT con las frecuencias de baja a media, que son los que tienen la mayor varianza, extraídos mediante un recorrido en zigzag, como se representa en la Figura 5. El coeficiente DCT del borde superior izquierdo es eliminado, ya que solo representa el valor promedio de las intensidades en cada punto.

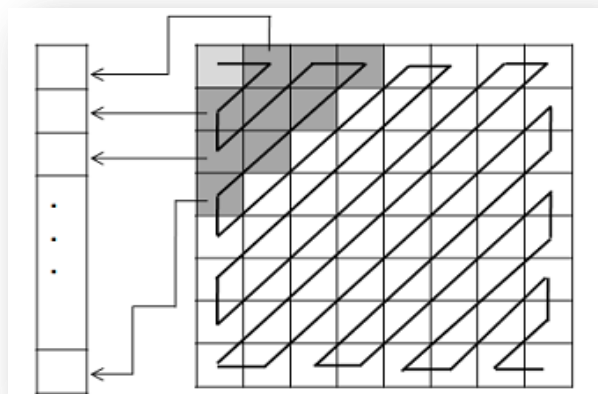


Figura 6. Representación de la extracción de rasgos DCT utilizando recorrido en zigzag (19).



La diferencia entre los métodos globales y locales radica en que, en el primer caso se halla la DCT para la imagen completa, se conforma el vector con los coeficientes seleccionados y con este se hace la clasificación, mientras que en el caso de los métodos locales primeramente se divide la imagen en bloques y luego se calcula la DCT y se construye el vector de coeficiente para cada uno de estos. En este último caso, una vez obtenidos los vectores por regiones, la fusión de estos puede hacerse mediante dos vías, una es conformar un vector concatenando los coeficientes DCT obtenidos en cada bloque y con este realizar la clasificación (fusión de características) y la otra es realizar la clasificación por separado en cada bloque combinando luego los resultados de la clasificación individual (fusión por decisión) (20).

El principal problema de este método radica en cuáles y cuántos coeficientes escoger para representar de manera efectiva los rasgos discriminatorios de un rostro y a la vez lograr robustez ante las distintas variaciones que pueden afectar la imagen. Como se mencionó, normalmente, se utilizan los coeficientes DCT asociados a las bajas frecuencias para conformar los vectores de rasgos, estos concentran la mayor información de la imagen y han permitido obtener resultados satisfactorios. Sin embargo, es también en estos coeficientes donde se reflejan las variaciones de iluminación sobre el rostro. En la Figura 7 se muestran dos imágenes de una misma persona y sus respectivas DCT.

Como puede observarse en la Figura 7, las variaciones en las condiciones de iluminación inciden directamente sobre los coeficientes DCT de bajas frecuencias, ubicados en la esquina superior izquierda de las imágenes que representan la DCT. A simple vista, las DCT de ambas imágenes parecen muy similares, pero cuando se hace un acercamiento se puede notar con más claridad que los coeficientes DCT de bajas frecuencias contienen información diferente y por tanto, al utilizarlos como representación de las imágenes de ese rostro, no permiten identificar que corresponden a la misma persona.

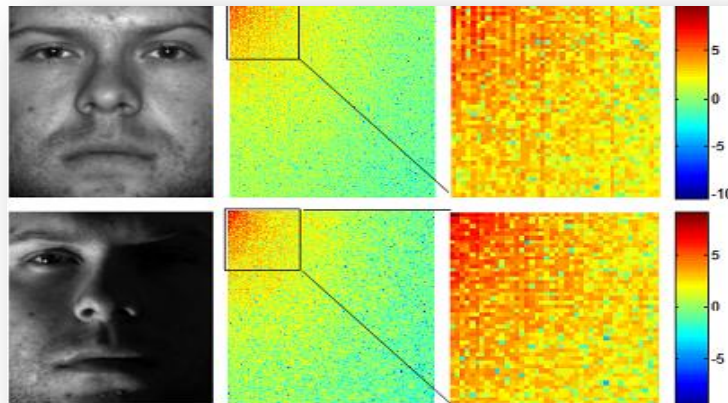


Figura 7. Influencia de las variaciones de iluminación en los coeficientes DCT de bajas frecuencias de dos imágenes de un mismo rostro (19).

En (11) se confirma que a pesar de sus ventajas, esta técnica no ofrece buenas propiedades de discriminación cuando se eliminan coeficientes buscando invarianza a la iluminación.

#### 1.5.2.2.4 Transformada Discreta de Wavelet (DWT)

La transformada wavelet, también conocida como transformada de ondula, es una clase de transformada de Fourier. Esta transformada es una forma de representación simultánea de tiempo y frecuencia. Estas wavelets (y las transformadas) pueden ser continuas o discretas. La transformada continua se utiliza en física, mientras que la discreta se utiliza en el análisis de señales, encontrando un campo de aplicación en ingeniería informática en análisis y compresión de imágenes digitales.

La Transformada Discreta de Wavelets (DWT) permite la descomposición de la imagen en diferentes escalas de coeficientes a la vez que se preserva la información.

En la descomposición se obtienen tanto coeficientes de aproximación como de detalle. Sobre los coeficientes de aproximación cabe, a su vez, aplicar nuevas transformaciones para conseguir un nivel inferior de resolución (21).

El análisis Wavelet representa el paso lógico siguiente a la STFT: es una técnica mediante ventanas con regiones de tamaño variable.

Una forma sencilla de comprender el modo de operación de esta Transformada, es considerando el siguiente ejemplo: Supóngase que se tiene una señal que posee frecuencias hasta 1000 [Hz<sup>5</sup>].

En la primera etapa se divide la señal en 2 partes, pasándola por un filtro pasa altos y uno pasa bajos. De este modo, se obtienen 2 diferentes versiones de la señal original: parte de la señal que corresponde al rango 0-500 [Hz] y la otra en el rango 500-1000 [Hz]. Posteriormente, se toma una de estas partes (o ambas) y se repite el proceso. Suponiendo que se hace nuevamente este proceso con la parte de baja frecuencia de la señal, y que en dicho resultado el proceso se vuelve a repetir en la parte de baja frecuencia, se tienen 4 partes de la señal original: 0-125 [Hz], 125-250 [Hz], 250-500 [Hz] y 500-1000 [Hz]. Se puede continuar con este procedimiento hasta que se haya descompuesto la señal en un determinado número de niveles. De este modo, se tienen un grupo de señales que representan a la misma señal, pero todas ellas corresponden a diferentes bandas de frecuencias (17).

El uso de esta transformada muestra muy buenos resultados como se demuestran en (14), en la cual se realizan experimentos, que manifiestan, que las diferencias en imágenes de diferentes personas son mayores que la diferencia de pose, iluminación y expresión en imágenes de una misma persona.

La extracción de características mediante la transformada wavelet discreta (DWT) permite reducir la dimensionalidad de las imágenes y al mismo tiempo suavizar las variaciones en la apariencia facial debido a cambios de iluminación, expresión facial y postura.

El algoritmo DWT es una técnica de ventaneo con regiones de tamaño variable, la cual permite descomponer una señal o imagen en versiones escaladas y trasladadas de una wavelet madre.

Calcular coeficientes wavelets para todas las posibles escalas requiere de una gran cantidad de cálculos y genera una cantidad enorme de información. Si únicamente se elige un subconjunto de escalas y posiciones, por ejemplo, muestras tomadas en potencias de 2 (escalas y posiciones dinámicas), entonces nuestro análisis será mucho más eficiente, esto se realiza mediante la DWT.

Una forma eficiente de implementar la DWT usando filtros fue desarrollada por Mallat<sup>6</sup> (22). Formalmente, este algoritmo se conoce como transformada rápida wavelet. El método propuesto por Mallat se lleva a cabo de la siguiente manera: en primer lugar, la señal se pasa a través de un filtro

---

<sup>5</sup> Hz: Unidad de frecuencia del Sistema Internacional de Unidades.

<sup>6</sup> G. Mallat profesor que hizo contribuciones al desarrollo de la teoría de wavelets, realizó trabajos en las matemáticas aplicadas, procesamiento de señales y la segmentación de la imagen.

paso-bajo y otro paso-alto simultáneamente, y posteriormente se realiza la operación de decimación ([Ver Anexo 1](#)) a la salida de cada filtro para eliminar los datos que no son relevantes. La DWT permite descomponer una señal o imagen en aproximaciones y detalles.

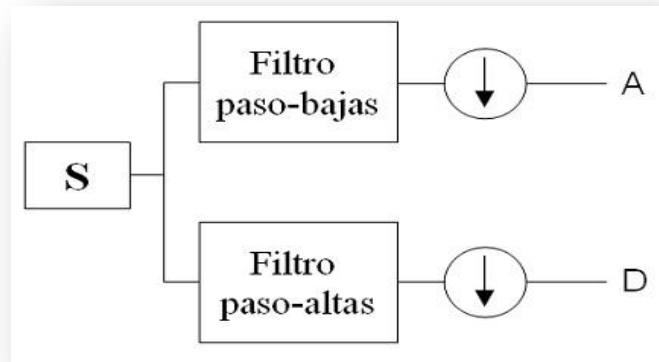


Figura 8. Esquema de filtrado para calcular la DWT unidimensional (14).

El esquema de filtrado propuesto por Mallat (22) para implementar la DWT unidimensional se muestra en la Figura 8. Los coeficientes de aproximación (A) se obtienen de la salida del filtro paso-bajas, mientras que los coeficientes de detalle (D) son proporcionados por el filtro paso-alto.

La descomposición mediante la DWT puede ser iterativa, esto se logra filtrando los coeficientes de aproximación en forma recursiva. A esto se le denomina árbol de descomposición. En cada proceso de filtrado se genera un nivel de descomposición. En la Figura 9 se ilustra el proceso de descomposición en tres niveles, en cada nivel se producen los respectivos coeficientes de aproximación y detalle. Conforme mayor sea el nivel de descomposición, el número de coeficientes es menor y su espectro corresponde a bandas de frecuencia más bajas. En teoría, la descomposición puede continuar indefinidamente, sin embargo, en la práctica es indispensable elegir un número de niveles de descomposición adecuado a nuestras necesidades.

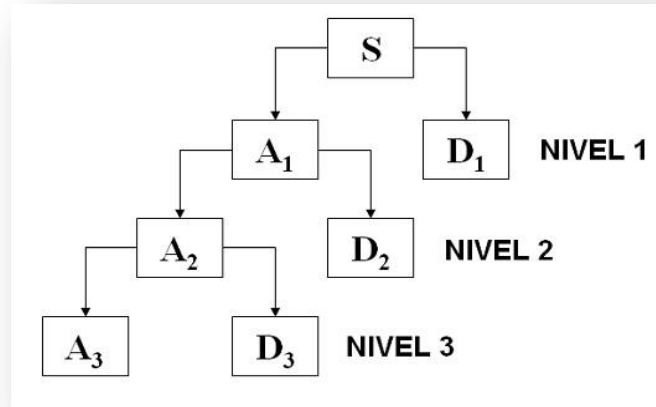


Figura 9. Árbol de descomposición por wavelets en tres niveles (14).

Para el caso bidimensional, la DWT puede calcularse aplicando la DWT unidimensional a las filas y posteriormente a las columnas. En la Figura 10 se ilustra el esquema de filtrado para implementar la DWT de una imagen.

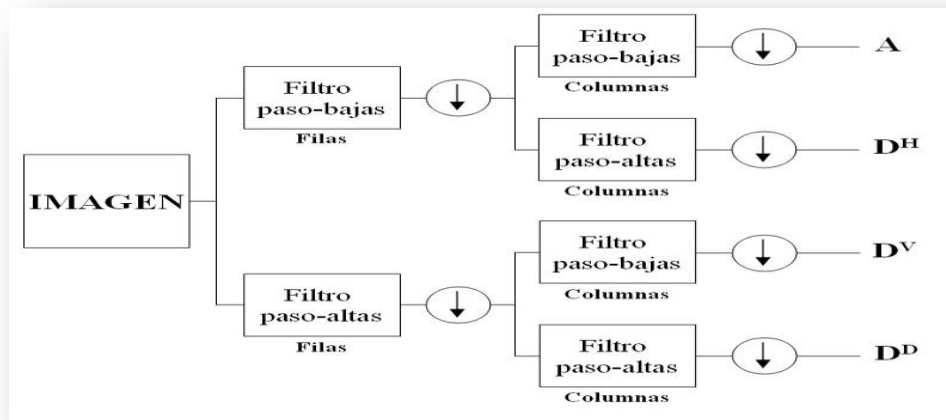


Figura 10. Esquema de filtrado para calcular la DWT de una imagen (14).

La aplicación de la DWT a una imagen produce cuatro matrices de coeficientes wavelet, una de aproximación y tres de detalles (horizontal, vertical y diagonal). La matriz de coeficientes de aproximación contiene los componentes en bandas de baja frecuencia de la imagen original mientras que las matrices de detalles los componentes en bandas de alta frecuencia. En la Figura 11 se

presenta la descomposición de la imagen facial en sus matrices de coeficientes de aproximación (A), detalles horizontales (H), detalles verticales (V) y detalles diagonales (D).

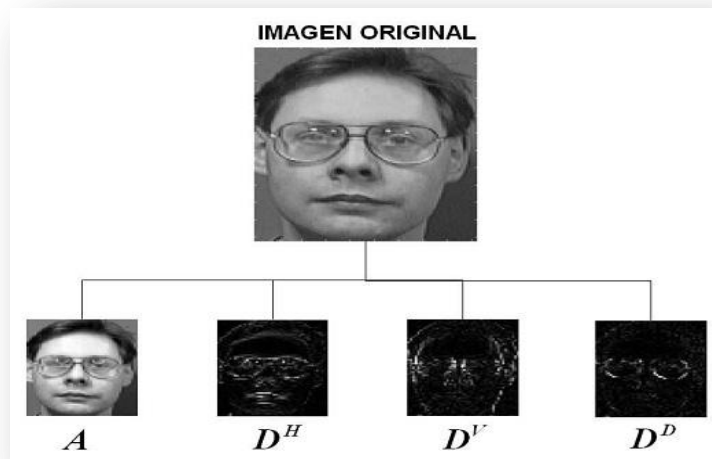


Figura 11. Descomposición de la imagen facial en matrices de aproximación y detalles (14).

Los componentes en bandas de baja frecuencia contribuyen a la descripción global de la imagen, la cual contiene la información más relevante para la diferenciación entre clases. Por tal motivo, la extracción de características se lleva a cabo mediante la descomposición recursiva del rostro, de donde los coeficientes en las bandas de más baja frecuencia son extraídos para formar los vectores característicos.

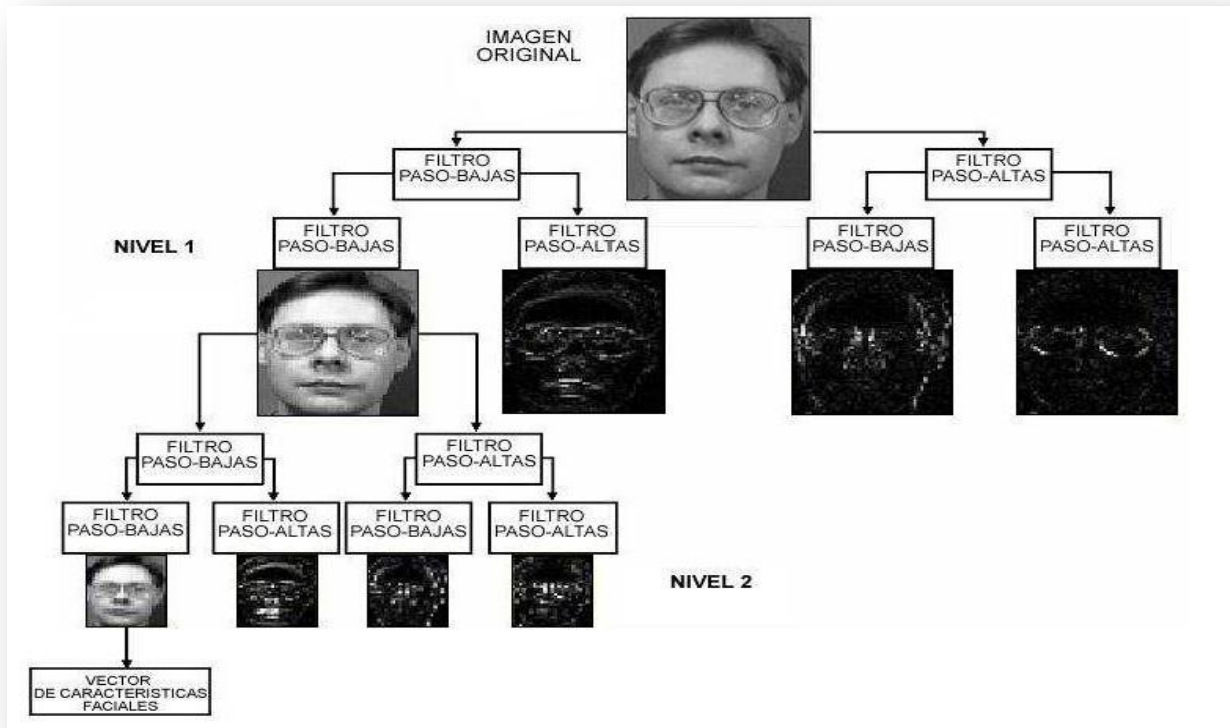


Figura 12. Esquema de extracción de características mediante la DWT (14).

La Figura 12 se muestra la descomposición en 2 niveles de un rostro, el vector característico es derivado de los componentes de baja frecuencia del segundo nivel de descomposición. La descomposición recursiva de la imagen facial conduce a la reducción de las dimensiones en un cuarto de su tamaño por cada nivel de descomposición.

Para demostrar que la DWT puede extraer características faciales estables y adecuadas para la diferenciación entre individuos, en (14) se realiza la evaluación de los vectores característicos derivados de la descomposición de los rostros.

En (14) se toman los vectores característicos de 10 imágenes de dos personas con variaciones en la apariencia facial y se puede apreciar con facilidad que las variaciones entre vectores característicos interclase son mayores que las variaciones intraclase por cambios de iluminación, expresión facial y postura.

#### 1.5.2.2.4.1 Familia de funciones Wavelet

La elección de la familia wavelet más adecuada para la descomposición de un rostro juega un papel muy importante en la obtención de una representación wavelet más robusta. Existen muchas familias de funciones wavelet que han mostrado su gran utilidad en diversas aplicaciones, entre ellas destacan: *Haar*, *Daubechies*, *Meyer*, *Biortogonal*, *Coiflet*, *Symlet*, reversa biortogonal. Describas en la ayuda del Matlab.

Haar: es la primera y más simple de todas las wavelets. Es una wavelet continua que se asemeja a una función escalón unitario.

Entre las más populares se encuentra la propuesta por *Daubechies* (23). Aunque estas funciones no son simétricas, para un orden lo suficientemente alto ( $N > 15$ ) resultan óptimas. Los nombres de la familia de *Daubechies* se escriben como dbN, db el sobrenombre y N es el orden, un entero positivo que denota el número de coeficientes de filtro que posee la wavelet.

Otra familia interesante es la que se define en Meyer (14), con una construcción basada en el espacio de Fourier. Esta función no es compacta.

Symlets: son wavelets generalmente escritas como symN, donde N es el orden y sym es el sobrenombre. Esta familia está compuesta por funciones wavelet casi simétricas propuestas por *Daubechies* como extensión de la familia db (14). Las propiedades de la familia de *Daubechies* y *Symlets* son muy similares.

Las familias de wavelets de orden superior suelen ser generalmente funciones suaves, sin picos fuertemente acentuados, y suelen proporcionar buena localización tanto en el espacio como en la frecuencia.

### 1.5.3 Análisis de los algoritmos estudiados

En general puede decirse que existe una gran cantidad de métodos desarrollados para la extracción de rasgos, por lo que es muy difícil decidir cuál algoritmo es el mejor para la extracción de características. Algunos métodos son más eficientes que otros en cuanto a variaciones de expresión, pero disminuyen su eficacia cuando se pretende analizar imágenes con variaciones en la iluminación. Por lo que algunos métodos son más robustos a ciertas variaciones que otros, pero más sensibles a otras. Por esta razón las teorías existentes han sido combinadas entre ellas buscando métodos más eficaces.



Con la DCT es difícil saber cuántos y cuáles coeficientes escoger para representar el vector característico ya que estos pueden estar afectados por la iluminación. La transformada de Fourier trabaja bien si la señal está compuesta de unos cuantos componentes estacionarios. Sin embargo, algún cambio repentino en el tiempo, en una señal no estacionaria, es separado del eje de frecuencias. Para contrarrestar esta desventaja se ha modificado la transformada de Fourier dando origen a lo que es la STFT también conocida como la transformada de Gabor. Esta última, aplica una función utilizada como ventana de análisis que se mueve sobre los datos para determinar el espectro. La desventaja que presenta esta transformada es que la ventana de análisis es de un tamaño fijo y por eso no puede adaptarse a las características de la señal en ciertos puntos interesantes de analizar.

Por lo visto anteriormente es que ha cobrado tanta fuerza la teoría de Wavelets y se dice que es el paso lógico siguiente a la STFT: una técnica mediante ventanas con regiones de tamaño variable.

Se decide utilizar la DWT para el desarrollo del componente porque es un método de extracción en el dominio de la frecuencia, donde es más fácil separar los efectos de la iluminación. También se aprovechan las características de la transformada wavelet para obtener vectores característicos más robustos a cambios de iluminación, expresión facial y postura. Además, la eliminación de los componentes de más alta frecuencia puede reducir la cantidad de información redundante en una imagen facial y al mismo tiempo se preserva la información necesaria para distinguir la identidad de dicho rostro (14).

Para fortalecer un poco más el proceso se combina el resultado de Wavelet con PCA. El poder de PCA proviene de su facilidad de cómputo y aplicabilidad general. Es un método muy común que se utiliza para reducir la dimensionalidad de un vector. El mismo hace un análisis global de los datos hallando la varianza total del sistema, para luego encontrar una representación en un espacio de menor dimensión que conserva la mayor cantidad de varianza posible (13).

Esto permitirá que los algoritmos de clasificación tengan un menor costo computacional y por tanto una mayor eficiencia; así como el ahorro de espacio físico que ocupan en disco y en memoria.

## 1.6 Metodologías, Lenguajes y Herramientas de Desarrollo

El entorno de desarrollo, metodología, lenguaje de programación y modelado, así como herramienta CASE, definido para la construcción del componente, fue producto de un estudio realizado por el equipo de desarrollo. Se brindará una breve descripción de cada herramienta a utilizar.

### 1.6.1 Metodologías de desarrollo de software

Todo desarrollo de *software* es riesgoso y difícil de controlar, siéndolo aún más, cuando no se siga una guía que pauté los pasos a seguir y muestre cómo actuar ante las dificultades. Este papel lo juegan las metodologías de desarrollo de *software*. Sin ellas, el resultado de un proyecto sería desastroso, con clientes insatisfechos con el resultado, malas planificaciones con fechas límite imposibles de cumplir y desarrolladores aún más insatisfechos.

Dada la variedad de metodologías que existe en la actualidad y la clasificación de las mismas, son muchos los factores a tener en cuenta a la hora de seleccionar una metodología para que esta se ajuste al ambiente de desarrollo y ayude, en vez de entorpecer, a obtener los resultados esperados.

Basándose en este planteamiento, se hizo un estudio de las mismas, para encontrar cuál se adapta mejor a una aplicación de pequeña escala con poco personal y requisitos cambiantes. Para ello se analizaron RUP (Proceso Racional Unificado) y XP.

#### 1.6.1.1 RUP

La metodología de desarrollo RUP forma parte de las llamadas metodologías pesadas. Entre sus principales características figuran que cuenta con una forma disciplinada de asignar tareas y responsabilidades, pretende implementar mejores prácticas, propone un desarrollo iterativo y el uso de la arquitectura basada en componentes además de un riguroso control de cambios, se basa en el modelado visual y le da gran importancia a la verificación de la calidad del *software*. El proceso de ciclo de vida de RUP se divide en cuatro fases llamadas Inicio, Elaboración, Construcción y Transición. Estas fases se dividen en iteraciones, cada una de las cuales produce una pieza de *software* demostrable Figura 13.

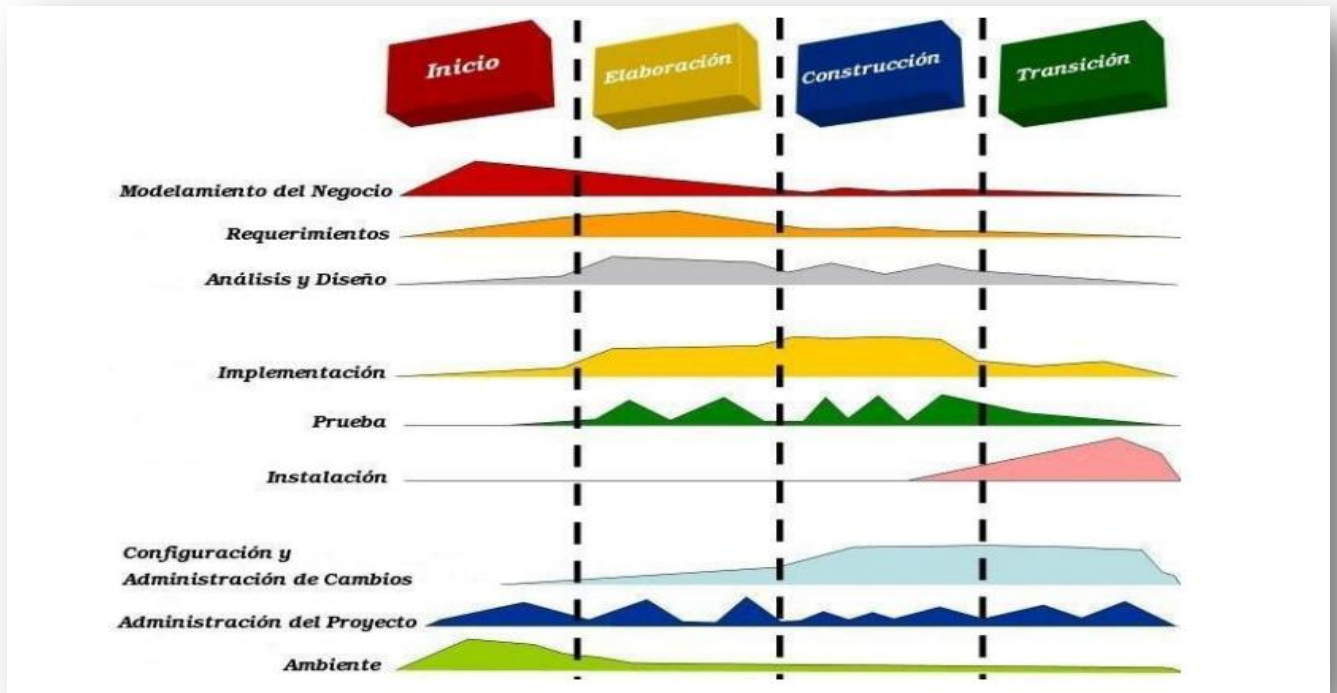


Figura 13. Fases del Procesos Unificado de RUP (24).

RUP es una metodología poco factible para ser utilizada en proyectos pequeños, pues puede que no sea posible cubrir los costos de dedicación del equipo de desarrollo, ya que tendrían que invertir una gran cantidad de horas y esfuerzo para hacer las cosas según lo que dicen los procesos.

### 1.6.1.2 XP

La programación extrema (XP) se basa en la simplicidad, la comunicación y el reciclado continuo de código. Promueve los valores de comunicación directa entre las personas; el coraje de exponer sus dudas, miedos, experiencias sin "embellecer" estas de ninguna de las maneras; el intento de mantener el *software* lo más simple posible y la capacidad de respuesta ante los cambios que se van haciendo necesarios a lo largo del camino. Los objetivos de XP son muy simples: la satisfacción del cliente. Esta metodología trata de dar al cliente el *software* que él necesita y cuando lo necesita. Por tanto, se debe responder muy rápido a las necesidades del cliente, incluso cuando los cambios sean al final del ciclo de la programación (25). El segundo objetivo es potenciar al máximo el trabajo en grupo. Tanto los jefes de proyecto, los clientes y desarrolladores, son parte del equipo y están involucrados en el desarrollo del *software*.

El ciclo de vida de XP se enfatiza en el carácter iterativo e incremental del desarrollo. Sus iteraciones son relativamente cortas ya que se piensa que cuanto más rápido se le entreguen resultados de valor al cliente, más retroalimentación se va a obtener, lo cual representará una mejor calidad del producto a largo plazo. El mismo se encuentra representado en la Figura 14.

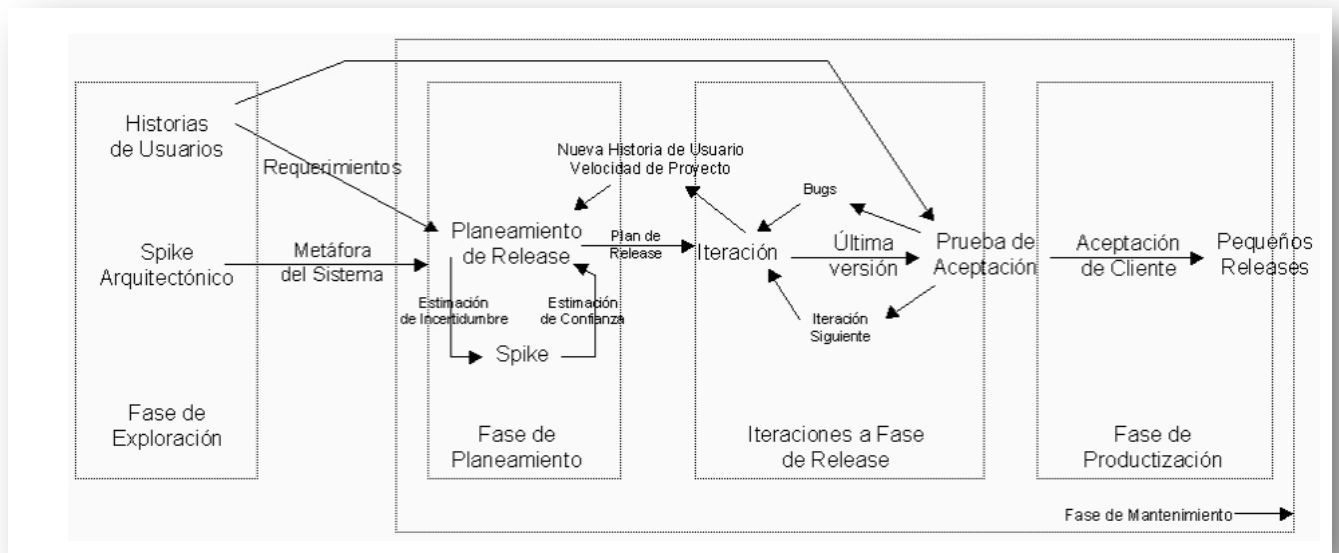


Figura 14. Representación de la metodología XP (26).

### 1.6.1.3 Selección de la metodología

Para la selección de la metodología de desarrollo a utilizar, se hace necesario analizar el entorno en el que se construirá el producto.

En primer lugar, es necesario destacar que los usuarios finales son los propios desarrolladores del departamento de Biometría, pues son los que tienen la necesidad de un componente que los abstraiga de toda la lógica relacionada con la extracción de las características de una imagen facial, lo cual significa que en todo momento se estará interactuando con los mismos.

Otro de los factores es que los requisitos asociados a la extracción de características pueden sufrir cambios debido al propio proceso de desarrollo del departamento de Biometría, lo cual arrojará nuevas necesidades y por ende nuevas funcionalidades a incluir en el componente.

Además, se hace necesario una solución rápida, pues el componente que se propone formará parte de los cimientos del sistema de identificación del departamento de Biometría y por lo tanto deberá estar funcional desde el momento en que dicho proyecto inicie su proceso de desarrollo.

En adición a lo antes descrito se cuenta con poco personal de desarrollo, lo cual dificulta la adopción de roles en cada etapa así como la generación de múltiples artefactos.

Una vez que el componente sea insertado al proceso de reconocimiento facial del departamento de Biometría, el mismo será sometido a constantes pruebas por parte de los desarrolladores, lo cual significa que se le estará dando mantenimiento al código a la vez que se desarrollen el resto de las iteraciones identificadas.

Se adoptaron los valores de XP comunicación, sencillez, retroalimentación, coraje y respeto. La comunicación entre el equipo de desarrollo y los clientes será maximizada y efectuada de forma directa e interpersonal con el fin de evitar los problemas y errores causados por la mala adopción de la misma. Se realizará un trabajo tan simple como sea posible, favoreciendo la comunicación, reduciendo el código sin utilizar y garantizando la calidad. Se estará constantemente midiendo el sistema para conocer cuánto se acerca a las funcionalidades necesarias mediante pruebas por parte de los usuarios. Los valores antes mencionados contribuirán a enfrentar con coraje cada cambio en los requisitos y refactorizaciones al sistema.

Lo antes descrito evidencia cómo las características del componente a desarrollar se inclinan a adoptar las prácticas y valores de la metodología XP, por lo cual se seleccionó la misma como rectora del proceso de desarrollo del *software*.

### **1.6.2 Lenguaje de Modelado**

El Lenguaje Unificado de Modelado (UML) por sus siglas en inglés, es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de *software*. Describe lo que supuestamente hará el sistema, pero no dice cómo implementarlo (27). Las herramientas CASE de modelado con UML permiten representar el *software* mediante diagramas que se generan durante las diferentes etapas del proyecto.

### 1.6.3 Herramientas de Modelado

#### 1.6.3.1 Herramienta CASE

Dentro de las herramientas claves en el desarrollo de aplicaciones informáticas se encuentran las herramientas de Ingeniería de *Software* Asistida por Ordenador (CASE) por sus siglas en inglés, las cuales son las encargadas de ayudar en el ciclo de desarrollo, con el fin de aumentar la productividad y reduciendo el costo en términos de tiempo y dinero. En el ciclo de desarrollo pueden ayudar en el proceso de diseño del proyecto, en el cálculo de costes, pueden implementar una parte del código, compilación automática y documentación (28). En el presente trabajo se utilizó Visual Paradigm para UML.

##### 1.6.3.1.1 Visual Paradigm para UML

Una de las herramientas CASE más usadas es la suite creada por Visual Paradigm International (VPI). VPI es un proveedor de soluciones informáticas que incluye organizaciones para desarrollar aplicaciones de calidad, rápidas y baratas. Está compuesta por productos que facilitan a las organizaciones la visualización y diseño de diagramas. Sus soluciones se enfocan en eliminar la complejidad, aumentando así la productividad y disminuyendo el tiempo de desarrollo de las aplicaciones informáticas (29).

Aunque Visual Paradigm para UML (VP) en su versión 8.0 Enterprise Edition es una herramienta muy completa y permite la representación de varios artefactos, no se aprovechan todas sus capacidades entre las que se encuentra la generación de código debido a que no genera código para el lenguaje de programación M o lenguaje de programación Matlab como suele llamarse en algunas bibliografías, el cual será utilizado para la implementación del componente. Por tal motivo VP se utiliza para la realización del Diagrama de componentes que sirve para un mejor entendimiento del funcionamiento y relación entre los archivos del componente desarrollado.

### 1.6.4 Lenguaje de Programación

Un lenguaje de programación es un conjunto de símbolos, reglas sintácticas y semánticas que definen su estructura, así como el significado de sus elementos y expresiones. Es utilizado para controlar el comportamiento físico y lógico de una máquina. En el presente trabajo se utilizó el lenguaje Matlab.

#### 1.6.4.1 Lenguaje Matlab

Es un lenguaje de alto nivel para matrices con sentencias para control de flujo, creación de funciones y estructuras de datos, funciones de entrada/salida y algunas características de programación orientada a objetos. Este lenguaje permite tanto la programación a pequeña escala para la creación rápida de programas, como programación a larga escala para la realización de aplicaciones complejas (30).

Como lenguaje es muy sencillo basado en vectores, arreglos y matrices, esto hace que la expresión de coeficientes de filtros y demoras de *buffers* sean muy simples de expresar y comprender. En los últimos años se ha añadido la extensión para programación orientada a objetos y el diseño de interfaces gráficas. Este lenguaje es propio de la herramienta Matlab.

#### 1.6.5 Entorno Integrado de Desarrollo

Un Entorno Integrado de Desarrollo (IDE), por sus siglas en inglés, es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica.

Teniendo en cuenta que MatLab es un programa interactivo para computación numérica y visualización de datos y está basado en un sofisticado *software* de matrices para el análisis de sistemas de ecuaciones, orientado para llevar a cabo proyectos en donde se encuentren implicados elevados cálculos matemáticos y la visualización gráfica de los mismos, se considera como IDE a utilizar para la implementación del componente (31).

Matlab integra análisis numérico, cálculo matricial, proceso de señal y visualización gráfica en un entorno completo donde los problemas y sus soluciones son expresados del mismo modo en que se escribirían tradicionalmente, sin necesidad de hacer uso de la programación adicional (31).

El nombre de Matlab proviene de la construcción de los términos *Matrix Laboratory* y fue inicialmente concebido para proporcionar fácil acceso a las librerías *Linpack* y *Eispack*, las cuales representan hoy en día dos de las librerías más importantes en computación y cálculo matricial (31).

Matlab es un sistema de trabajo interactivo cuyo elemento básico de trabajo son las matrices. El programa permite realizar de modo rápido la resolución numérica de problemas en un tiempo mucho menor que si se quisiesen resolver estos mismos problemas con lenguajes de programación

tradicionales como por ejemplo: Fortran, Basic y C (31).

Matlab goza en la actualidad de un alto nivel de implantación en escuelas y centros universitarios, así como en departamentos de investigación y desarrollo de muchas compañías industriales nacionales e internacionales. En entornos universitarios, por ejemplo, Matlab se ha convertido en una herramienta básica, tanto para los profesionales e investigadores de centros docentes como una importante herramienta para la impartición de cursos universitarios, tales como sistemas e ingeniería de control, álgebra lineal, proceso digital de imagen, señal, etc. En el mundo industrial, Matlab está siendo utilizado como herramienta de investigación para la resolución de complejos problemas planteados en la realización y aplicación de modelos matemáticos en ingeniería. Los usos más característicos de la herramienta se encuentran en áreas de computación y cálculo numérico tradicional, prototipaje algorítmico, teoría de control automático, estadística, análisis de series temporales para el proceso digital de señal.

Matlab dispone también en la actualidad de un amplio abanico de programas de apoyo especializado, denominado *Toolboxes*, que extienden significativamente el número de funciones incorporadas en el programa principal. Estos *Toolboxes* cubren en la actualidad prácticamente casi todas las áreas principales en el mundo de la ingeniería y la simulación, destacando entre ellos el “*toolbox*” de proceso de imágenes, señal, control robusto, estadística, análisis financiero, matemáticas simbólicas, redes neuronales, lógica difusa, identificación de sistemas, simulación de sistemas dinámicos, etc (31).

Matlab emplea matrices porque con ellas se puede describir infinidad de sucesos de una forma altamente flexible y matemáticamente eficiente. Una matriz de píxeles puede ser una imagen o una película. Una matriz de fluctuaciones de una señal puede ser un sonido o una voz humana. Y tal vez más significativamente, una matriz puede describir una relación lineal entre los componentes de un modelo matemático. En este último sentido, una matriz puede describir el comportamiento de un sistema extremadamente complejo. Por ejemplo una matriz puede representar el vuelo de un avión a 40 000 pies de altura, o un filtro digital de procesamiento de señales (31).



### **1.7 Conclusiones Parciales**

En el presente capítulo se plantearon conceptos fundamentales para lograr una mejor comprensión de la problemática, así como las temáticas que permiten el desarrollo de la investigación. Se describe la metodología que regirá el desarrollo del componente de extracción de características faciales y se abordaron las tecnologías y herramientas seleccionadas para el desarrollo del componente.

Con el estudio realizado se sentaron las bases que permitirán la agilización del desarrollo del componente de extracción de características faciales seleccionando la herramienta CASE Visual Paradigm, el IDE Matlab R2008a. XP será de suma importancia para guiar el proceso de desarrollo del componente respectivamente.

# Capítulo 2

## ANÁLISIS Y DISEÑO

### 2.1 Introducción

La solución a desarrollar, debe ser fruto de un correcto análisis y una amplia comprensión de todos los elementos que se relacionan con el tema de la extracción de características en imágenes faciales, profundizándose en el estudio de las características que posibilitan desarrollar el mismo.

El objetivo que se persigue con la elaboración de este capítulo es mostrar la evolución de la solución durante las fases iniciales de Exploración y Planificación, además de presentar los diferentes artefactos generados en las mismas, los cuales serán premisas cruciales para la entrega final del componente.

En este capítulo se interpretan las necesidades del sistema especificándolas mediante historias de usuario y requisitos no funcionales. Se expone la estimación de cada una de ellas, se define el tiempo que va a demorar cada iteración que permitirá la realización del componente, así como también se da a conocer el plan de entrega del componente.

### 2.2 Exploración

Para comenzar el desarrollo de un producto de *software* es necesario asegurarse de que es posible hacerlo. Se debe tener confianza en que las herramientas seleccionadas ayudarán a la culminación del trabajo. Se debe creer que una vez que el código se haga, este puede utilizarse cada día. Cada miembro del equipo debe confiar en sus propias habilidades y en las habilidades de los otros.

La fase de exploración ayuda a resolver todos estos conflictos. Es en ella donde los clientes plantean a grandes rasgos las historias de usuario que son de interés para el producto. Durante la misma, los desarrolladores interactúan con las herramientas a utilizar durante todo el proceso, exploran activamente las posibilidades de arquitectura y experimentan los límites de las tecnologías a utilizar (25).

### 2.2.1 Descripción del sistema

Para el funcionamiento del componente el usuario debe indicar el directorio donde se encuentra la base de datos de entrenamiento y la imagen de prueba a las que se les extraerá el vector de características. La base de datos de entrenamiento es la que contiene imágenes faciales de diferentes personas y la imagen de prueba es una imagen facial de la persona que se desea reconocer o comprobar si existe en la base de datos antes mencionada.

Si las imágenes cargadas son en colores, deben ser llevadas a escala de grises para un mejor procesamiento de las mismas. En caso de que estén procesadas no es necesario llevarlas al nivel de gris.

Culminado este proceso las imágenes están listas para aplicarles la descomposición con el algoritmo de wavelet. Se debe seleccionar el tipo de wavelet y los niveles de descomposición a los que se van a someter las imágenes. Una vez obtenidos los vectores de características invariantes de cada imagen se procede aplicarles el algoritmo PCA para reducir la dimensionalidad de dicho vector, ya que este obtiene de cada imagen solo los principales componentes, procedimiento que reduce el costo computacional en la próxima etapa que es la de clasificación.

Conseguido el vector de características de todas las imágenes, se pueden exportar a un fichero .xml para su posterior análisis. Estarán separados en 2 archivos diferentes el vector de la imagen de prueba y los de la base de datos de entrenamiento.

### 2.2.2 Arquitectura de la solución

El patrón de arquitectura propuesto para estructurar el componente de extracción de características faciales es el de Tuberías y Filtros, que por lo general es apropiada para sistemas que implementan transformaciones de datos en pasos sucesivos.

La estructura que provee para procesar flujos de datos la arquitectura Tuberías y Filtros, es que cada paso de procesamiento se encapsula en un filtro, donde los datos se pasan usando los tubos entre filtros adyacentes. Recombinado los filtros se pueden construir distintas familias de sistemas relacionados. Para el componente, el flujo de datos se comporta de la siguiente manera con la entrada de una imagen. La misma pasará por un filtro para extraer características utilizando el componente WaveletExtraccion.m, del mismo se filtrará el vector característico extraído para que pase por otro tubo hacia el componente PCAReducion.m para obtener una menor dimensión del

vector característico. Dicho vector puede filtrarse por el componente DistanciaEuclidiana.m para obtener la imagen que más se asemeje a la imagen de entrada.



Figura 15 Arquitectura Tuberías y Filtros

### 2.2.3 Usuarios del Sistema

Los actores del sistema no son parte de él. Pueden intercambiar información con él. Pueden ser un recipiente pasivo de información. Pueden representar el rol que juega una o varias personas, un equipo o un sistema automatizado.

A continuación se muestra el único actor identificado.

Tabla 1. Actor del sistema.

Actor del Sistema	Descripción
Desarrollador	Representa a la persona que interactúa con el componente para cargar las imágenes a procesar y extraer los vectores característicos de las mismas.

### 2.2.4 Historias de Usuario

Las historias de usuario (HU) representan las funcionalidades que el cliente desea que estén presentes en el sistema; por lo tanto, todo el trabajo futuro debe girar en torno a satisfacer estas expectativas. Las historias de usuario son la primera y fundamental entrada para el proceso de desarrollo de *software* basado en la metodología XP. La descripción de las mismas se muestra a continuación:

Tabla 2. Transformar imágenes a escala de grises.

Historia de Usuario	
<b>Número:</b> HU1	<b>Nombre de Historia de Usuario:</b> Transformar imágenes a escala de grises.
<b>Modificación de Historia de Usuario:</b> Ninguna	
<b>Usuario:</b> Eider Barnet Díaz y Anet González Jarrosay	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Alto	<b>Puntos Reales:</b> 1
<b>Descripción:</b> En caso de que las imágenes a analizar sean en color, se debe transformar la imagen a escala de grises para un mejor procesamiento. El procedimiento consiste en transformar los colores en tonalidades de grises que van desde el negro hasta el blanco. Una explicación más amplia del procedimiento se encuentra en el <a href="#">anexo 3</a> .	
<b>Observaciones:</b> En caso de que las imágenes a analizar sean procesadas, no es necesario transformar la imagen a escala de grises.	

Tabla 3. Aplicar transformada discreta de Wavelet.

Historia de Usuario	
<b>Número:</b> HU2	<b>Nombre de Historia de Usuario:</b> Aplicar transformada discreta de Wavelet.
<b>Modificación de Historia de Usuario:</b> Ninguna	
<b>Usuario:</b> Eider Barnet Díaz y Anet González Jarrosay	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 2

<b>Riesgo en Desarrollo:</b> Alto	<b>Puntos Reales:</b> 2
<b>Descripción:</b> Luego de cargar las imágenes, seleccionar el tipo de wavelet y el nivel de descomposición para extraer el vector de características de cada una de ellas. Este algoritmo utiliza filtros de paso alto y paso bajo y se repite según la cantidad de niveles de descomposición escogidos. Se obtienen como resultado el vector de cada imagen.	
<b>Observaciones:</b>	

Tabla 4. Aplicar análisis de componentes principales (PCA).

Historia de Usuario	
<b>Número:</b> HU3	<b>Nombre de Historia de Usuario:</b> Aplicar el análisis de los componentes principales (PCA).
<b>Modificación de Historia de Usuario:</b> Ninguna	
<b>Usuario:</b> Eider Barnet Díaz y Anet González Jarrosay	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 3
<b>Riesgo en Desarrollo:</b> Alto	<b>Puntos Reales:</b> 3
<b>Descripción:</b> Luego de tener el vector característico de cada imagen se le aplica el algoritmo PCA para obtener una representación más reducida del vector característico.	
<b>Observaciones:</b>	

Tabla 5. Exportar vector característico.

Historia de Usuario	
<b>Número:</b> HU3	<b>Nombre de Historia de Usuario:</b> Exportar el vector característico.
<b>Modificación de Historia de Usuario:</b> Ninguna	
<b>Usuario:</b> Eider Barnet Díaz y Anet González Jarrosay	<b>Iteración Asignada:</b> 2
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Medio	<b>Puntos Reales:</b> 1

<b>Descripción:</b> Luego de obtener los vectores de características, estos se pueden guardar en un fichero .xml <sup>7</sup> para su posterior análisis. Se debe indicar la dirección específica donde los desea guardar.
<b>Observaciones:</b>

Tabla 6. Calcular distancia euclidiana.

Historia de Usuario	
<b>Número:</b> HU3	<b>Nombre de Historia de Usuario:</b> Calcular distancia euclidiana
<b>Modificación de Historia de Usuario:</b> Ninguna	
<b>Usuario:</b> Eider Barnet Díaz y Anet González Jarrosay	<b>Iteración Asignada:</b> 2
<b>Prioridad en Negocio:</b> Media	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 1
<b>Descripción:</b> Se halla la distancia euclidiana desde cada vector de características de las imágenes de la base de datos al vector de la imagen que se quiere reconocer. La que menor distancia posea será la que mejor describe las características de la imagen de la persona que se quiere reconocer.	
<b>Observaciones:</b> El método de la distancia euclidiana es utilizado en la fase de clasificación pero se utiliza para comprobar que los vectores extraídos son adecuados para el reconocimiento.	

### 2.2.5 Especificación de los requisitos de software

La obtención de los requisitos es un paso muy importante para las siguientes etapas del desarrollo del *software*, pues un error en estas fases iniciales puede dar al traste con un sistema que no cumpla las expectativas de los usuarios y difícilmente aporte valor agregado al negocio para el que debe ser concebido. La IEEE (*Institute of Electrical and Electronics Engineers*) define un requisito como condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo. El flujo de trabajo de requisitos tiene como propósito fundamental guiar el desarrollo hacia el sistema correcto. Esto se consigue mediante una descripción de los requisitos del sistema (es decir, las condiciones o capacidades que el sistema debe cumplir) suficientemente buena como para que pueda llegarse a un acuerdo entre el cliente (incluyendo a los usuarios) y los desarrolladores sobre qué debe y qué no debe hacer el sistema (24).

<sup>7</sup>XML: Lenguaje de marcado extensible

### 2.2.6.1 Requisitos no funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido y confiable, por ejemplo, pudiera pensarse que el sistema responda dentro de un intervalo de tiempo especificado o que obtenga los resultados de los cálculos con un nivel de precisión dado. En muchos casos los requisitos no funcionales son fundamentales para el éxito del producto. Normalmente están vinculados a requisitos funcionales, es decir, una vez que se conozca lo que el sistema debe hacer se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grave debe ser (32).

Los requisitos no funcionales del sistema no se capturan en la metodología XP en forma de historias de usuario, puesto que los clientes generalmente no conocen la terminología técnica que se utiliza para describir este tipo de requisitos. La captura de requisitos no funcionales es por tanto una tarea del equipo de desarrollo completo, que haciendo un intercambio profundo de ideas con el cliente, determinan las mejores soluciones para crear un sistema que, además de implementar correctamente las historias de usuario, cumpla con determinados estándares de calidad y con las características propias del negocio que se pretende automatizar (33).

A continuación se definen varios requisitos no funcionales con los que debe cumplir el componente para su desarrollo lo que permitirá un mayor rendimiento, seguridad y calidad del mismo.

#### 2.2.6.1.1 Software

- El sistema será usado en Sistema Operativo Windows XP o superior.

#### 2.2.6.1.2 Requisitos de hardware mínimo

- PC Intel Pentium 4 o superior.
- CPU 3.0 GHz o superior.
- 256 MB de Memoria RAM o superior.

#### 2.2.6.1.3 Rendimiento

- La aplicación está concebida para un ambiente donde la misma se integre a un sistema de identificación, por lo que los tiempos de respuestas deben ser rápidos (menores que 6 segundos), así como la velocidad de procesamiento de información.



#### 2.2.6.1.4 Requisitos de seguridad

- **Confiabilidad:** El componente propuesto debe ser capaz de adaptarse con el fin de alcanzar un equilibrio interno frente a los cambios externos del entorno.

### 2.3 Planificación

Se planea garantizar que siempre se esté haciendo lo más importante que queda por hacer, a fin de mantenerse coordinados de manera efectiva con otras personas y para responder rápidamente a eventos inesperados (25).

El propósito de la fase de planificación es establecer un acuerdo entre los clientes y desarrolladores sobre el menor tiempo en que la mayor cantidad de HU pueda ser realizada.

Para ello se realizó la planificación con el objetivo de maximizar el valor del *software* producido a partir de la puesta en producción de las características más importantes lo antes posible, se realiza una estimación del esfuerzo requerido para la implementación de las HU y se define el tiempo de entrega y cada iteración.

#### 2.3.1 Estimación de esfuerzo por Historias de Usuario

A continuación se muestra la estimación del esfuerzo por cada HU propuesta para el desarrollo de la aplicación:

Tabla 7. Estimación de esfuerzo por Historias de Usuario.

No	Historias de Usuario	Estimación (Semanas)
1.	Transformar imágenes a escala de grises.	1
2.	Aplicar transformada discreta de Wavelet.	2
3.	Aplicar el análisis de los componentes principales (PCA).	3
4.	Exportar el vector característico.	1
5.	Calcular distancia euclidiana.	1

### 2.3.2 Plan de Iteraciones

Después de ser identificadas y descritas las HU y estimar el esfuerzo dedicado a la realización de cada una de ellas, se procede a la planificación de la fase de implementación estableciendo una división de dos iteraciones.

**Iteración 1:** En esta iteración se realizarán las historias de usuarios más complejas y que inciden críticamente en la lógica de la aplicación. Una vez concluida esta iteración el componente se encontrará en un estado plenamente funcional.

**Iteración 2:** En esta iteración se realizarán las historias de usuarios que son menos complejas y que no inciden críticamente en la lógica de la aplicación. Una vez concluida esta iteración el componente se encontrará completamente concluido y listo para su explotación.

#### 2.3.2.1 Plan de duración de las iteraciones

El plan de duración de las iteraciones se encarga de mostrar las HU en el orden en que se implementarán en cada iteración así como la duración estimada de las mismas.

A continuación, el plan de duración de las iteraciones que permite el desarrollo del componente:

Tabla 8. Plan de duración de las iteraciones.

Iteración	Código	Historias de Usuarios	Duración estimada (semanas)
Iteración 1	HU1	Transformar imágenes a escala grises	6
	HU2	Aplicar transformada discreta de Wavelet	
	HU3	Aplicar el análisis de los componentes principales (PCA)	
Iteración 2	HU4	Exportar el vector característico	2
	HU5	Calcular distancia euclidiana	

- La Iteración 1 durará en total 1 mes y dos semanas.
- La Iteración 2 durará en total 2 semanas.

### 2.3.3 Plan de entregas

Una vez que se concluye la tarea por parte del cliente de elaborar las distintas HU, se comienza con la creación del Plan de entrega, para estimar el tiempo de desarrollo de las mismas. Este artefacto se elabora con la intención de fijar qué período de tiempo puede tardar la implementación de cada una de las historias en cada iteración, definiéndose las fechas en que serán liberadas las versiones funcionales del producto.

Plan de entrega del componente de extracción de características es:

Tabla 9. Plan de entrega.

1ra Entrega (Iteración 1)	2da Entrega (Iteración 2)
13 de Mayo 2012	27 de Mayo 2012

### 2.4 Conclusiones Parciales

En este capítulo se realizaron las diferentes HU, se definió la estimación de esfuerzo por HU permitiendo así obtener un plan de duración y un plan de entrega con la realización del componente. La arquitectura de Tuberías y Filtros utilizada, brinda la posibilidad de ir transformando un flujo de datos en un proceso comprendido por varias fases secuenciales siendo la entrada de cada una la salida de la anterior.

# Capítulo 3

## IMPLEMENTACIÓN Y PRUEBA

### 3.1 Introducción

La implementación de *software* en XP es un proceso que se realiza de forma iterativa, obteniendo como resultado un producto funcional que debe ser sometido a pruebas y mostrado al cliente para permitir una retroalimentación por parte de los desarrolladores. El siguiente capítulo está dedicado a detallar las 2 iteraciones llevadas a cabo durante la etapa de construcción del componente, exponiéndose cada una de las tareas designadas por HU. De igual modo se exponen las pruebas de aceptación efectuadas sobre el componente.

### 3.2 Fase de Diseño.

Siguiendo las guías de la metodología XP, debe aparecer un diseño que describa qué clases hay y cómo interactúan. La metodología XP no requiere la representación del sistema mediante diagramas de clases utilizando el Lenguaje Unificado de Modelado (UML); en su lugar usa una técnica para representar clases, esta es la que usa tarjeta Clase-Responsabilidad-Colaboración (CRC). No obstante, el uso de diagramas UML puede aplicarse siempre y cuando influyan en el mejoramiento de la comunicación, no sea un peso su mantenimiento, no sean extensos y se enfoquen en la información importante (33).

Programar en Matlab es usar una serie de comandos que permitan realizar una tarea o función específica. Estos pueden ser escritos uno por uno a través de la línea de comandos:

#### Archivos -M: Comandos y Funciones:

Los archivos de disco que contienen instrucciones de Matlab se llaman archivos-M. Esto es así porque siempre tienen una extensión de ".m" como la última parte de su nombre de archivo.

Un archivo -M consiste de una secuencia de instrucciones normales de Matlab, que probablemente incluyen referencias a otros archivos-M. Un archivo -M se puede llamar a sí mismo recursivamente. Se pueden crear archivos-M utilizando un editor de texto o procesador de palabras.

Hay dos tipos de archivos -M: los de comandos y las funciones. Los archivos de comandos, automatizan secuencias largas de comandos. Los archivos de funciones, permiten añadir a Matlab funciones adicionales expandiendo así la capacidad de este programa. Ambos, comandos y funciones, son archivos ordinarios de texto ASCII.

### Archivos de Comandos

Cuando un archivo de comandos es invocado, Matlab simplemente ejecuta los comandos encontrados en dicho archivo. Las instrucciones en un archivo de comando operan globalmente en los datos en el espacio de trabajo. Los comandos son utilizados para hacer análisis, resolver problemas, o diseñar secuencias largas de comandos que se conviertan en interactivas.

### Archivos de Funciones

Un archivo -M que contiene la palabra *function* al principio de la primera línea, es un archivo de función. En una función, a diferencia de un comando, se deben de pasar los argumentos. Las variables definidas y manipuladas dentro de la función son locales a esta y no operan globalmente en el espacio de trabajo. Los archivos de funciones se utilizan para extender a Matlab y crear nuevas funciones para Matlab utilizando el lenguaje propio de Matlab.

Con lo antes mencionado utilizando la metodología XP es innecesario usar tarjetas CRC si se desarrolla con Matlab. La no utilización de estas está dada a que la implementación en Matlab es mediante comandos y funciones y no emplea clases. El programa se implementa mediante la invocación de funciones.

Para un mejor entendimiento del funcionamiento y relación entre los archivos del componente desarrollado se emplea un diagrama de componentes.

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Además, muestran la organización y las dependencias lógicas entre un conjunto de componentes de *software*. Para la construcción del componente se modeló el siguiente diagrama.

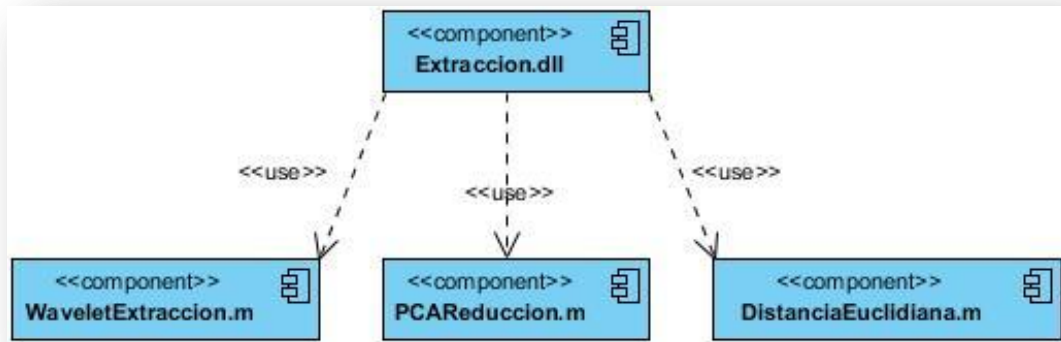


Figura 16. Diagrama de componente.

### 3.2.1 Funciones implementadas para el desarrollo del componente

El componente desarrollado como parte de la solución de esta investigación es una Biblioteca de enlace dinámico, Dll (*Dynamic Link Library*) por sus siglas en inglés. Dicho componente cuenta con una serie de funcionalidades distribuidas en varios subcomponentes. Ver Figura 16.

Subcomponente WaveletExtraccion.m contiene la implementación de la función:

#### **WaveletExtraccion (UbicacionBDEntrenamiento, UbicacionImagenPrueba, NivelDescomposicion, TipoWavelet, SiColor)**

Esta función es la que le aplica la DWT a todas las imágenes. Se le pasa por parámetros la ruta o dirección de la base de datos de entrenamiento y la imagen de prueba, además del nivel de descomposición y el tipo o familia de wavelet a utilizar. El valor del parámetro SiColor es 0 en caso que las imágenes sean procesadas y 1 en caso que sean en color. El propósito de esta última variable es para transformar las imágenes a escala de grises en caso que estas sean en color. Los resultados de esta función serán 2 variables, una será el vector característico de la imagen de prueba y en la otra estarán los vectores de cada imagen de la base de datos. El vector de cada imagen será devuelto en una columna, es decir, se concatena una columna debajo de la otra.

Subcomponente PCAReducion.m contiene la implementación de la función:

### **PCAReducion (ImágenesEntrenamiento, ImagenPrueba)**

Esta función es la que reduce la dimensión de los vectores característicos. Recibe como parámetros la salida de la función WaveletExtraccion, es decir, los vectores de la base de datos y el de la imagen de prueba. Esta función retorna los vectores con una dimensión menor.

Subcomponente DistanciaEuclidiana.m contiene la implementación de la función:

### **DistanciaEuclidiana (ImágenesProyectadas, ImagenPruebaProyectada)**

Esta función calcula la distancia euclidiana que existe entre cada imagen de la base de datos a la imagen de prueba. A la misma se le debe pasar por parámetros ImágenesProyectadas que son los vectores de la base de datos e ImagenPruebaProyectada que será el vector de la imagen de prueba. El retorno será el índice o el número de la imagen dentro de la base de datos de menor distancia, por ejemplo si es la tercera imagen, devuelve el número 3.

La solución también permitirá guardar los vectores de características utilizando la función:

### **GuardarVector (LugarAGuardarVector, VectorCaracteristico)**

Esta función se encarga de guardar los vectores. El parámetro VectorCaracteristico puede ser el vector de la imagen de prueba o los vectores de la base de datos. En el parámetro LugarAGuardarVector se debe especificar la ruta completa con el nombre seleccionado y la extensión, por ejemplo C:/Users/Administrador/Desktop/Procesadas/vectorBD.xml.

## **3.2.2 Principales funciones utilizadas del Matlab**

**dir ('dirname');** retorna una lista de los archivos almacenados en el directorio especificado en 'dirname'.

**Size (X, dim);** retorna el tamaño de la dimensión de 'X' especificada en 'dim'.

**strcmp ('str1', 'str2');** compara los *strings* str1 y str2 y retorna lógico 1 (verdadero) si son idénticos, y retorna lógico 0 (falso).

**int2str (N);** convierte el entero 'N' en *string*.

**strcat (s1, s2, s3,...);** concatena los *string* horizontalmente.

**Imread (filename);** lee la imagen del archivo especificado.

**rgb2gray (RGB);** convierte una imagen en RGB a escala de grises para eliminar el color y la información de saturación.

**wavedec2 (X, N,'wname');** descomposición bidimensional de la imagen 'X' a 'N' niveles con el tipo de wavelet 'wname'.

**appcoef2 (C,S,'wname', N);** devuelve los coeficientes de aproximación, es decir, el vector de características de la imagen, al nivel 'N' con la wavelet 'wname' pasándole los argumentos coeficientes de todas las componentes 'C' y el tamaño de cada componente 'S'.

**detcoef2 (O, C, S, N);** devuelve de la descomposición wavelet los coeficientes de detalle horizontal, vertical y diagonal especificado en 'o'. Parar O = 'h'(o 'v' o 'd', respectivamente).

**Reshape (A, m, n);** cambia la forma de una matriz, hace una matriz A de mXn.

**Mean (A, dim);** devuelve el promedio o valor medio de los elementos de la dimensión 'dim' de A.

**[V, D] = eig(A);** devuelve matrices de eigenvectores 'V' y eigenvalores 'D' de 'A'. Tal que  $A \cdot V = V \cdot D$ . La matriz D es la forma canónica de A, donde los eigenvalores de A están en la diagonal principal. La matriz V es la matriz modal, sus columnas son los eigenvectores de A.

**uint8 (x);** convierte x en tipo de datos entero de 8 bits.

**Double (x);** convierte x en tipo de datos double.

### 3.2.3 Estándar de programación

Es necesario establecer un criterio fijo que proporcione reglas para la creación de nombre para variables y métodos del tal forma que ninguno de los integrantes del equipo dudase en ningún momento del nombre que debiera poner a un método o del nombre que tiene que utilizar para llamar a un método desarrollado por otros programadores del equipo.

Por esto es necesario establecer un estándar de codificación de tal forma que los programadores sigan los mismos criterios para desarrollar el código.



Es decisivo, para poder plantear con éxito la propiedad colectiva del código. Esta sería impensable sin una codificación basada en estándares que haga que todo el mundo se sienta cómodo con el código escrito por cualquier otro miembro del equipo (34).

#### Definición de ciclos

Utilizar constantes con nombre en lugar de números, como por ejemplo en lugar de emplear For i = 1 To 7 se debe utilizar For i = 1 To NumeroImágenes para que resulten fáciles de comprender.

#### Comentarios

Encima de cada función debe ir un comentario que explique el propósito de la misma para un mejor entendimiento.

Alinear todos los comentarios de final de línea en la misma posición de tabulación.

#### Sangría

En las líneas de construcción lógica debe ser de 4 espacios, por ejemplo:

```
If expression1
```

```
    statements1
```

```
Else if expression2
```

```
    statements2
```

```
Else
```

```
    statements3
```

```
End
```

#### Espacios

Utilizar un espacio antes y después de cada operador.

Usar un espacio luego de una coma.

No usar espacios entre las variables o nombre de función y los corchetes o paréntesis.

Nombres de Funciones

Todas las palabras deben comenzar con letra mayúscula y separadas por un guión bajo.

Nombre de Variables

Todas las palabras deben comenzar con letras mayúsculas y unidas una de otras.

Nombre de Componentes visuales

La primera palabra define el tipo de componente y comienza con minúscula, las siguientes van unidas a la primera y comienzan con letra mayúscula.

**3.2.4 Tareas de la Ingeniería**

XP plantea que la implementación de un *software* se hace iterativamente, obteniendo al culminar cada iteración un producto funcional, que debe ser probado y mostrado al cliente. Durante el transcurso de las iteraciones, se realiza la implementación de las HU definidas por el cliente y descritas por el equipo de desarrollo en la etapa de Exploración. Como parte de este plan, se descomponen estas HU en tareas de la ingeniería las cuales son asignadas a los programadores para ser implementadas durante la iteración correspondiente.

A continuación se describen las tareas correspondientes a las historias de usuario.

Tabla 10. HU1\_Transformar imágenes a escala de grises y mostrar la imagen de prueba transformada.

Tarea de ingeniería	
<b>Número tarea:</b> 1	<b>Número historia de usuario:</b> HU1
<b>Nombre tarea:</b> Transformar imágenes a escala de grises.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha inicio:</b> 02/04/2012	<b>Fecha fin:</b> 08/04/2012
<b>Programador responsable:</b> Anet González Jarrosay	
<b>Descripción:</b> <ul style="list-style-type: none"><li>• Desarrollar el proceso de transformar a escalas de grises las imágenes de la base de datos de entrenamiento y la imagen de prueba.</li></ul>	

Tabla 11. HU2\_Desarrollar el algoritmo de la transformada discreta de Wavelet.

Tarea de ingeniería	
Número tarea: 2	Número historia de usuario: HU2
Nombre tarea: Desarrollar el algoritmo de la transformada discreta de Wavelet.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 09/04/2012	Fecha fin: 22/04/2012
Programadores responsables: Eider Barnet Díaz y Anet González Jarrosay	
<b>Descripción:</b> <ul style="list-style-type: none"> <li>Desarrollar la transformada discreta de Wavelet para extraer los vectores característicos de las imágenes.</li> </ul>	

Tabla 12. HU3\_Desarrollar el algoritmo análisis de componentes principales (PCA).

Tarea de ingeniería	
Número tarea: 3	Número historia de usuario: HU3
Nombre tarea: Desarrollar el algoritmo análisis de componentes principales (PCA).	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 23/04/2012	Fecha fin: 13/05/2012
Programadores responsables: Eider Barnet Díaz y Anet González Jarrosay	
<b>Descripción:</b> <ul style="list-style-type: none"> <li>Desarrollar el algoritmo PCA para reducir la dimensión de los vectores característicos extraídos por wavelet.</li> </ul>	

Tabla 13. HU4\_Exportar vector característico.

Tarea de ingeniería	
Número tarea: 4	Número historia de usuario: HU4
Nombre tarea: Exportar los vectores característicos.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 14/05/2012	Fecha fin: 20/05/2012
Programador responsable: Eider Barnet Díaz	

**Descripción:**

- Desarrollar la funcionalidad de guardar los vectores característicos en un fichero .xml para su posterior análisis.

Tabla 14. HU5\_Calcular distancia euclidiana.

<b>Tarea de ingeniería</b>	
<b>Número tarea:</b> 5	<b>Número historia de usuario:</b> HU5
<b>Nombre tarea:</b> Calcular distancia euclidiana.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha inicio:</b> 21/05/2012	<b>Fecha fin:</b> 27/05/2012
<b>Programador responsable:</b> Eider Barnet Díaz	
<b>Descripción:</b>	
<ul style="list-style-type: none"> <li>• Desarrollar el algoritmo de la distancia euclidiana para comprobar que los vectores característicos extraídos son adecuados para el reconocimiento.</li> </ul>	

### 3.2.5 Requisitos para la integración del componente

Una vez implementados los algoritmos se decide crear el componente para la plataforma .NET de Microsoft. La misma se selecciona debido a la experiencia que tienen los desarrolladores del departamento de Biometría en el trabajo con esta tecnología, además de ser la definida para el trabajo en dicho departamento.

Una de las características más interesantes de .NET es que brinda la posibilidad de elegir el lenguaje de programación de las aplicaciones entre un gran abanico que incluye lenguajes populares como C#, C++, *Visual Basic*, *Perl*, *Python*, etc.

El componente fue realizado con interfaz gráfica de usuario de la herramienta de despliegue de Matlab, se puede ejecutar mediante el comando "*deploytool*"

Para un correcto despliegue se debe tener en cuenta:

- La máquina donde se vaya a utilizar debe tener el mismo sistema operativo que la máquina donde se compiló la dll.
- Garantizar que el compilador de Matlab (Matlab Compiler Runtime (MCR)) esté instalado en la máquina y que sea la versión 7.13, si el MCR no está instalado debe ejecutar MCRInstaller.exe. En caso de tener el Matlab instalado se puede localizar en:  
C:\Program Files\MATLAB\R2010a\toolbox\compiler\deploy\win32\MCRInstaller.exe

En el directorio raíz del proyecto deben estar los archivos:

- Extraccion.ctf (contiene código-M compilado)
- Extraccion.dll (componente o solución)
- ManagedCPPAPI.netmodule (módulo necesario por el compilador .NET) se puede encontrar en C:\Program Files\MATLAB\R2010a\toolbox\dotnetbuilder\bin\[win32|Win64]v2.0

- MWArray.dll. Esta librería puede ser encontrada en:

C:\Program Files\MATLAB\R2010a\toolbox\dotnetbuilder\bin\[win32|Win64]v2.0, provee clases que actúan como puente entre la aplicación y el MCR. La librería Extracción.dll solamente trabajará con una MWArray.dll construida con la misma versión del MCR. La clase MWArray provee métodos para crear y acceder a tipos de datos usados por el MCR. Incluye conversiones de clases como MWNumericArray, MWLogicalArray, MWCharArray, MWStructArray, and MWCellArray. Cada clase tiene propiedades y métodos que pueden ser usadas para saber varios atributos de las clases como número de dimensiones, tamaño de dimensión, tamaño del elemento y nombre del campo. Esas clases proporcionan los constructores necesarios, métodos y operadores para la creación de arreglos.

La librería Extracción.dll, se puede asociar a una aplicación agregándola como referencia al proyecto de Visual Studio. Igualmente hay que agregar la librería MWArray.dll a las referencias.

Las librerías compiladas son imposibles modificarlas desde Visual Studio, pero se puede acceder a los métodos implementados como si se tratasen de una clase más del proyecto.

Si se realizan cambios en el código fuente de los algoritmos se debe volver a generar la librería y acceder a la nueva versión de la misma desde el entorno de desarrollo.

Para más información acerca del proceso de instalación y el MCR, puede escribir en la ventana de comandos del MATLAB "help MCR" o "help mcristaller".

### 3.3 Pruebas.

Las pruebas son acciones en las cuales el sistema es ejecutado bajo condiciones o requisitos determinados. Los resultados son chequeados y registrados. Las pruebas verifican los resultados de la implementación del sistema. El modelo de prueba indica cómo han de ser probados aspectos determinados del producto o *software*. Hay multitud de conceptos asociados a las tareas de prueba. Clasificarlas es difícil, pues no son mutuamente disjuntas, sino muy entrelazadas. En lo que sigue se intentará una pequeña organización de acuerdo con sus características.

#### 3.3.1 Pruebas de aceptación.

Las pruebas de aceptación son creadas a partir de las HU. Durante una iteración la HU seleccionada en la planificación de iteraciones conllevará una o varias pruebas de aceptación. El cliente o usuario especifica los aspectos a testear cuando una HU ha sido correctamente implementada.

Una HU puede tener más de una prueba de aceptación, tantas como sean necesarias para garantizar su correcto funcionamiento y no se considera completa hasta que no supera sus pruebas de aceptación. Es responsabilidad del cliente verificar la corrección de las pruebas y tomar decisiones acerca de las mismas.

Para las pruebas se utilizaron imágenes de la base de datos pública ORL, la cual se puede descargar de forma gratuita de (35). Los archivos se encuentran en formato PGM, con una resolución de 92x112 píxeles con 256 niveles de gris por píxel. ([Ver Anexo 2](#)).

Además, se usaron bases de datos de imágenes en color accesibles en el CISED.

Para la compresión y visualización de los procesos desarrollados en el componente y tomando en cuenta que la extensión del mismo es .dll, lo cual imposibilita observar por parte del equipo de desarrollo las transformaciones que ocurren en las imágenes, se decide implementar una aplicación visual en la herramienta Matlab. Dicha aplicación para su funcionamiento hará uso del componente desarrollado. Esto facilitará en gran medida el trabajo de los desarrolladores, ya que a través de la

aplicación visual, se podrán realizar los procesos de cargar imágenes, aplicar los algoritmos Wavelet y PCA, así como, la distancia euclidiana, esta última utilizada para comprobar si el vector extraído es adecuado para efectuar los procesos posteriores (Clasificación).

Tabla 15. CP1\_Transformar las imágenes a escala de grises.

Caso de Prueba de Aceptación	
<b>Código de caso de prueba:</b> CP1	<b>Nombre de la historia de usuario:</b> Transformar las imágenes a escala de grises.
<b>Responsable de la prueba:</b> Anet González Jarrosay.	
<b>Descripción de la prueba:</b>	
<ul style="list-style-type: none"> <li>• Prueba de funcionalidad para transformar las imágenes a escala de grises.</li> </ul>	
<b>Condición de ejecución:</b>	
<ul style="list-style-type: none"> <li>• El usuario debe haber seleccionado en el menú buscar imágenes en color, en caso contrario no hace nada.</li> </ul>	
<b>Entrada/Pasos de ejecución:</b>	
<ul style="list-style-type: none"> <li>• El proceso se realiza automáticamente después de haber cargado las imágenes en color.</li> </ul>	
<b>Resultado esperado:</b>	
<ul style="list-style-type: none"> <li>• Todas las imágenes son transformadas a escala de grises.</li> <li>• Muestra la imagen de prueba transformada a escala de grises.</li> </ul>	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

Tabla 16. CP2\_Aplicar transformada discreta de Wavelet.

Caso de Prueba de Aceptación	
<b>Código de caso de prueba:</b> CP2	<b>Nombre de la historia de usuario:</b> Aplicar transformada discreta de Wavelet.
<b>Responsable de la prueba:</b> Eider Barnet Díaz.	
<b>Descripción de la prueba:</b>	
<ul style="list-style-type: none"> <li>• Prueba de funcionalidad para descomponer las imágenes mediante Wavelet. Cada imagen de la base de datos y la imagen de prueba son procesadas por Wavelet para extraer el vector de características de cada una de las imágenes.</li> </ul>	

<b>Condición de ejecución:</b> Ninguna.
<b>Entrada/Pasos de ejecución:</b> <ul style="list-style-type: none"> <li>El usuario debe seleccionar el tipo de wavelet y el nivel de descomposición que se va a utilizar para extraer el vector. Luego debe presionar el botón Aplicar Algoritmo.</li> </ul>
<b>Resultado esperado:</b> <ul style="list-style-type: none"> <li>Se extrae el vector característico de cada imagen.</li> <li>Se muestran los resultados después de aplicarle la transformada Wavelet a la imagen de prueba. Dicho resultado son 4 imágenes, la de aproximación, detalles horizontales, verticales y diagonales.</li> </ul>
<b>Evaluación de la prueba:</b> Prueba satisfactoria.

Tabla 17. CP3\_Aplicar el análisis de principales componentes (PCA).

Caso de Prueba de Aceptación	
<b>Código de caso de prueba:</b> CP3	<b>Nombre de la historia de usuario:</b> Aplicar el análisis de los principales componentes (PCA).
<b>Responsable de la prueba:</b> Anet González Jarrosay.	
<b>Descripción de la prueba:</b> <ul style="list-style-type: none"> <li>Prueba de funcionalidad para aplicar PCA a los resultados arrojados por el algoritmo Wavelet.</li> </ul>	
<b>Condición de ejecución:</b> Ninguna.	
<b>Entrada/Pasos de ejecución:</b> <ul style="list-style-type: none"> <li>Cada vector extraído por la Transformada de Wavelet es procesado seguidamente por PCA para reducir la dimensión del vector.</li> </ul>	
<b>Resultado esperado:</b> <ul style="list-style-type: none"> <li>Se extraen los vectores con los principales componentes de las imágenes.</li> <li>Se muestra como resultado del proceso 2 imágenes, una con el rostro promedio de la base de datos de entrenamiento y otra que es la imagen de diferencia entre la imagen de prueba y el rostro promedio.</li> </ul>	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	



Tabla 18. CP4\_Exportar vector característico.

Caso de Prueba de Aceptación	
<b>Código de caso de prueba:</b> CP4	<b>Nombre de la historia de usuario:</b> Exportar el vector característico.
<b>Responsable de la prueba:</b> Eider Barnet Díaz.	
<b>Descripción de la prueba:</b> <ul style="list-style-type: none"> <li>• Prueba de funcionalidad para guardar los vectores de características de las imágenes.</li> </ul>	
<b>Condición de ejecución:</b> Ninguna.	
<b>Entrada/Pasos de ejecución:</b> <ul style="list-style-type: none"> <li>• El usuario presiona el botón “Exportar Vector” luego de haber extraído los vectores de características mediante los algoritmos propuestos.</li> </ul>	
<b>Resultado esperado:</b> <ul style="list-style-type: none"> <li>• Se muestra una ventana donde el usuario debe seleccionar el directorio donde desea guardar los vectores de las imágenes.</li> <li>• Se crean 2 archivos de extensión .xml, uno con el vector de la imagen de prueba y otro con los vectores de cada imagen de la base de datos de entrenamiento en la ubicación seleccionada.</li> </ul>	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

Tabla 19. CP5\_Mostrar resultados del reconocimiento.

Caso de Prueba de Aceptación	
<b>Código de caso de prueba:</b> CP5	<b>Nombre de la historia de usuario:</b> Calcular distancia euclidiana.
<b>Responsable de la prueba:</b> Anet González Jarrosay.	
<b>Descripción de la prueba:</b> <ul style="list-style-type: none"> <li>• Prueba de funcionalidad para demostrar que los vectores extraídos son adecuados para el reconocimiento facial.</li> </ul>	
<b>Condición de ejecución:</b> Ninguna.	
<b>Entrada/Pasos de ejecución:</b> <ul style="list-style-type: none"> <li>• Se debe presionar el botón “Buscar imagen más cercana” para mostrar de los</li> </ul>	

rostros de la base de datos el más cercano a la imagen de prueba.
<b>Resultado esperado:</b> <ul style="list-style-type: none"> <li>• Se calcula la distancia euclidiana de cada rostro de la base de datos al rostro de prueba.</li> <li>• Se muestra la imagen de la base de datos de entrenamiento más cercana a la imagen de prueba.</li> </ul>
<b>Evaluación de la prueba:</b> Prueba satisfactoria.

Tabla 20. CP6\_Probar costo computacional de vectores wavelet y PCA.

Caso de Prueba de Aceptación	
<b>Código de caso de prueba:</b> CP6	<b>Nombre de la historia de usuario:</b> Calcular distancia euclidiana.
<b>Responsable de la prueba:</b> Eider Barnet Díaz.	
<b>Descripción de la prueba:</b> <ul style="list-style-type: none"> <li>• Prueba de funcionalidad para demostrar que el empleo de PCA mejora el tiempo de ejecución del proceso de clasificación.</li> <li>• Saber tiempo de ejecución de la distancia euclidiana con vectores característicos de Wavelet y PCA.</li> </ul>	
<b>Condición de ejecución:</b> Tener los vectores de características extraídos.	
<b>Entrada/Pasos de ejecución:</b> <ul style="list-style-type: none"> <li>• Se debe presionar el botón “Buscar imagen más cercana” para mostrar el rostro de la base de datos más cercano a la imagen de prueba.</li> </ul>	
<b>Resultado esperado:</b> <ul style="list-style-type: none"> <li>• El tiempo de ejecución para analizar los vectores resultantes de PCA debe ser menor que el de analizar los vectores resultantes de Wavelet.</li> </ul>	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

Tabla 21. CP7\_Familia wavelet.

Caso de Prueba de Aceptación	
<b>Código de caso de prueba:</b> CP7	<b>Nombre de la historia de usuario:</b> Aplicar transformada discreta de Wavelet.
<b>Responsable de la prueba:</b> Anet González Jarrosay.	
<b>Descripción de la prueba:</b>	
<ul style="list-style-type: none"> <li>• Prueba para saber cuál familia de Wavelet es mejor para el reconocimiento.</li> </ul>	
<b>Condición de ejecución:</b> Ninguna.	
<b>Entrada/Pasos de ejecución:</b>	
<ul style="list-style-type: none"> <li>• Se tomaron 6 imágenes de prueba al azar de la base de datos ORL. En la base de datos de entrenamiento había 9 imágenes en diferentes posturas y gestos de las mismas 6 personas escogidas. Se tuvo en cuenta el tiempo de ejecución del algoritmo con las familias de Wavelet <i>Daubechies</i> (db), <i>Symlets</i> (sym), <i>BiosSplines</i> (bior) y <i>Coiflets</i> (coif) al nivel 4 de descomposición como se muestra en la Figura 17.</li> </ul>	
<b>Resultado esperado:</b>	
<ul style="list-style-type: none"> <li>• Según (14) db y sym deben estar entre las mejores.</li> </ul>	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

Fallaron en el reconocimiento dos veces bior y una vez coif. Por lo que se concluye que las mejores familias para el reconocimiento de las escogidas son db y sym.

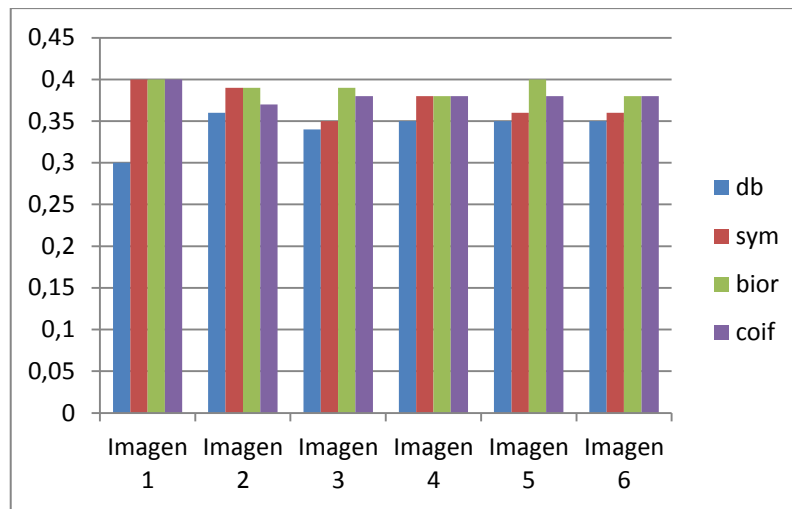


Figura 17. Familias de wavelet en el nivel 4 de descomposición.

Tabla 22. CP8\_Nivel de descomposición de wavelet

Caso de Prueba de Aceptación	
<b>Código de caso de prueba:</b> CP8	<b>Nombre de la historia de usuario:</b> Aplicar transformada discreta de Wavelet.
<b>Responsable de la prueba:</b> Eider Barnet Díaz.	
<b>Descripción de la prueba:</b>	
<ul style="list-style-type: none"> <li>Prueba para saber qué nivel de descomposición es mejor para el reconocimiento.</li> </ul>	
<b>Condición de ejecución:</b> Ninguna.	
<b>Entrada/Pasos de ejecución:</b>	
<ul style="list-style-type: none"> <li>Para saber el mejor nivel de descomposición se tomaron 5 imágenes de prueba al azar de la base de datos ORL. En la base de datos de entrenamiento habían 9 imágenes en diferentes posturas y gestos de las mismas 5 personas escogidas. Todas las imágenes fueron descompuestas en los niveles de descomposición del 1 al 4 con las familias de Wavelet de la prueba anterior.</li> </ul>	
<b>Resultado esperado:</b>	
<ul style="list-style-type: none"> <li>Se determina el mejor nivel de descomposición para el reconocimiento.</li> </ul>	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

Al descomponer recursivamente un rostro empleando la DWT, los detalles son removidos y las dimensiones reducidas, sin embargo, después de un determinado número de descomposiciones, es posible que los vectores característicos de distintas clases comiencen a aproximarse entre sí debido a que la representación wavelet del rostro tiende a ser más general. Como se puede observar en la Figura 18 en el primer y segundo nivel todas las familias reconocieron a las 5 personas pero ya en el tercer nivel comienzan a fallar *bior* y *coif*. Por lo que se concluye que el nivel más adecuado para la descomposición es el segundo.

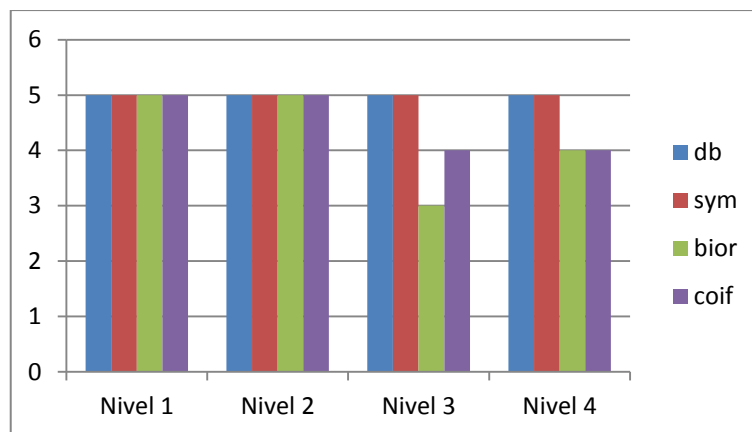


Figura 18. Familias de wavelet en niveles del 1 al 4.

### 3.4 Conclusiones Parciales

Luego de terminadas las iteraciones llevadas a cabo para la construcción del componente y la realización de las pruebas a cada una de las HU, se concluye que:

- El desglose de las historias de usuario en tareas de la ingeniería fue una buena práctica que mostró a los programadores las funcionalidades específicas a implementar.
- La utilización de estándares de programación permitió que los desarrolladores se sintieran más cómodos al implementar y siguieran una guía común para un mejor entendimiento del código.
- El empleo de las funcionalidades ya implementadas por Matlab, suavizó el trabajo del equipo de desarrollo, ya que se redujeron los cálculos y operaciones matemáticas.
- Las pruebas de aceptación concluyeron de manera exitosa demostrando la satisfacción del cliente con la solución.

## **CONCLUSIONES GENERALES**

- I. El componente para la extracción de características desarrollado muestra que la transformada wavelet discreta representa una buena forma de caracterizar un rostro para su reconocimiento.
- II. La representación wavelet hizo posible que las diferencias por cambios de identidad sean mayores que las diferencias intraclase producidas por cambios de iluminación, postura y expresión facial. Estas últimas en las imágenes de los rostros se atenúan cuando se realiza la descomposición mediante wavelets.
- III. De las familias de Wavelet utilizadas se comprobó que la mejor para la extracción de características era db y sym, en el nivel 2.
- IV. La Utilización del PCA permitió reducir las dimensiones de los vectores característicos, siendo de gran utilidad en la etapa de clasificación, ya que reduce el costo computacional del algoritmo a utilizar y por tanto permite una mayor eficiencia.

## **RECOMENDACIONES**

Como resultado de la investigación y el componente realizado, surgen nuevas ideas para trabajos futuros, las mismas se enuncian a continuación a modo de recomendaciones:

- I. Investigar nuevos métodos de extracción o combinaciones de ellos que permitan un mejor rendimiento.
- II. Integrar este componente a futuros sistemas de identificación que se desarrollen en el departamento de biometría del CISED.
- III. La extensión del componente a imágenes en tres dimensiones.

## REFERENCIA BIBLIOGRÁFICA

1. Li, Stan and Jain, A. ***Handbook of face recognition***. New York : Springer, 2005. ISBN 0-387-40595-X.
2. CENATAV. Sitio oficial de CENATAV. [Online] 2004. [Cited: Abril 12, 2012.] <http://www.cenatav.co.cu>.
3. Zhao, W, et al., et al. ***Face Recognition: A Literature Survey***. New York, USA : ACM Computing Surveys, 2003. pp. 399-458. Vol. 35. ISSN: 0360-0300.
4. Goldstein, A.J. ***Identification of human faces***. s.l. : Proceedings of the IEEE, 1971. pp. 748-760.
5. Brunelli, Roberto and Poggio. ***Face recognition: features versus templates***. Washington, USA : IEEE Transactions on Pattern Analysis and Machine Intelligence, 1993. ISSN: 0162-8828.
6. Beumer, G.M. ***A landmark paper in face recognition***. 2006 : Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition (FGR'06), 2006. ISBN:0-7695-2503-2.
7. Niu, Z, et al., et al. ***Enhance ASMs Based on AdaBoost-Based Salient Landmarks Localization and Confidence-Constraint Shape Modeling***. Berlín : Springer, 2005. ISBN:3-540-29431-7 978-3-540-29431-3.
8. Shi, J., Samal, A. and Marx, D. ***How effective are landmarks and their geometry for face recognition***. New York, USA : Computer Vision and Image Understanding, 2006. ISSN: 1077-3142.
9. Liu, D.H, Lam, K.M and Shen, L S. ***Illumination invariant face recognition***. New York : Pattern Recognition, 2005. ISSN: 0031-3203.
10. Chen, S, et al., et al. ***Face Recognition from a Single Image per Person: A Survey***. China : Pattern Recognition, 2006. ISSN: 0031-3203.
11. Saban, M and Sanderson, C. ***On Local Features for Face Verification***. Washington : IDIAP, 2005. ISBN:0-7695-2316-1.
12. Rizo Rodríguez, D, Méndez Vázquez, H and Garcia Reyes, E. ***Estado del arte de los métodos de extracción de rasgos invariantes a la iluminación en imágenes de rostros***. Ciudad de la Habana : s.n., 2011. ISSN 2072-6287.
13. Turk, M. ***Eigenfaces for Recognition***. Cambridge, USA : Journal of Cognitive Neuroscience, 1991. Vol. 3. ISSN: 0898-929X.
14. Fuentes Pineda, G. ***Reconocimiento de rostros mediante Wavelet y redes Neuronales***. Mexico : s.n., 2008.
15. Savvides, M, Vijaya Kumar, B. and Khosla, P. ***"Corefaces"- Robust Shift Invariant PCA based Correlation Filter for Illumination Tolerant Face Recognition***. Washington, USA : Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004.
16. Acharya, T and Ray, A.K. ***Image Processing: Principles and Applications***. Canada : s.n., 2005. ISBN-10: 0471719986.
17. Kouro, S and Musalem, R. ***Tutorial introductorio a la Teoría de Wavelet***. 2002.
18. González, C and Woods, R. ***Digital Image Processing***. s.l. : 3ra Edición. New Jersey. EEUU Prentice Hall, 2007. ISBN-10: 013168728X.
19. Méndez Vázquez, Heydi. ***Algoritmo de reconocimiento de rostros basados en la apariencia local para aplicaciones reales en condiciones variables de iluminación***. Ciudad de la Habana : s.n., 2012. ISSN 2072-6287.
20. Kemal Ekenel, H and Stiefelhagen, R. ***Local appearance based face recognition using discrete cosine transform***. Washington, USA : In 13th European Signal Processing Conference (EUSIPCO), 2005. ISBN:0-7695-2646-2.



21. Pajares, G and Guijarro, M. ***Fusión de imágenes mediante Waveletes para extracción de las partes verdes en imágenes agrícolas.*** 2011.
22. Mallat, G. ***A theory for multiresolution signal decomposition: the wavelet representation.*** New York : IEEE Transactions on Pattern Analysis and Machine Intelligence, 1989. ISSN 0162-8828.
23. Daubechies, J. ***Ten Lectures on Wavelets.*** s.l. : SIAM, 1992. ISBN 0898712742.
24. Jacobson, Ivar, Booch, Grady and Rumbaugh, James. ***El proceso unificado de desarrollo de software.*** s.l. : Pearson Addison Wesley, 2000. ISBN: 8478290362.
25. Beck, Kent. ***Extreme Programming Explained.*** s.l. : 2da Edición, 2004. ISBN-10: 0321278658.
26. Daniel Hernández, Dayana. ***Componente arquitectónico para la generación de reportes.*** Ciudad de la Habana : s.n., 2009.
27. Larmin, Craig. ***UML y Patrones. Introducción al análisis y diseño orientado a objetos.*** 2004. ISBN: 8420534382.
28. Ordoñez Leyva, Yoanni and Avilés Vázquez, Ernesto. ***Herramienta informática de Minería de Uso de la Web sobre los registros de navegación por Internet.*** Ciudad de la Habana : s.n., 2010.
29. Visual Paradigm. **Sitio Oficial de Visual Paradigm.** [Online] Noviembre 9, 2010. [Cited: Abril 20, 2012.] <http://www.visual-paradigm.com>.
30. Caldas, Francisco Jose. **Matlab.** [Online] [Cited: Mayo 6, 2012.] <http://gemini.udistrital.edu.co/comunidad/estudiantes/ocala/matlabTut/acerca.php>.
31. Docstoc. [Online] [Cited: Mayo 10, 2012.] <http://www.docstoc.com/docs/2147683/Tutorial-de-MatLab>.
32. Sommerville, Ian. ***Ingeniería del Software.*** Madrid : Pearson educación, 2005. ISBN: 84-7829-074-5.
33. Martínez Borrego, Kerlins. ***Implementación de un sistema para la gestión de eventos y servicios literarios en la Universidad de las Informáticas: UCILITERARIA.*** 2010.
34. Escribano, G.F. ***Extreme Programming.*** London : Springer-Verlag London, UK ©2002, 2002. ISBN:3-540-44024-0.
35. AT&T Laboratories Cambridge. **The orl database of faces.** [Online] 1994. [Cited: Mayo 15, 2012.] <http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>.
36. Méndez, Heydi, Rizo, Dayron and García, Edel. ***Estado del arte de los métodos de extracción de rasgos invariantes a la iluminación en imágenes de rostros.*** Ciudad de la Habana : s.n., 2011. ISSN 2072-6287.
37. Hernández, Roger. ***Estudio de técnicas de reconocimiento facial.*** Barcelona : s.n., 2010.
38. Rivas Perea, Pablo, Velarde Anaya, Omar and Valencia Olvera, Leonardo. ***Reconocimiento Facial en Ambientes No-Cooperativos Mediante Hough-KLT en Tiempo.*** 2006.
39. Cátedra: Visión Artificial. ***Comandos de MatLab utilizados para el procesamiento de imágenes.*** 2005.
40. Abdulwahhab Ghidan, Adil . ***Face Recognition Based On Mixed Between Selected Feature By Multiwavelet And Particle Swarm Optimization.*** Washington, USA : Developments in E-systems Engineering, 2010. ISBN: 978-0-7695-4160-0.
41. Alba, José Luis, Cid, Jesús and Mora, Inmaculada. ***Extracción de características.*** 2006.
42. Castrillo, William, Alvarez, Damian and López, Andrés. ***Técnicas de extracción de características en imágenes para el reconocimiento de expresiones faciales.*** 2008. ISSN 0122-1701.
43. Chun Lee, Yi and Hsing Chen, Chin. ***Feature Extraction for Face Recognition Based on Gabor Filters and Two Dimensional Locality Preserving Projections.*** Washington, USA : Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2009. ISBN: 978-0-7695-3762-7.

44. Conilione, Paul and Wang, Dianhui . ***Feature Extraction for Face Image Retrieval.*** Washington, USA : The 3rd International Conference on Innovative Computing Information and Control (ICIC'08), 2008. ISBN: 978-0-7695-3161-8.
45. Daugman, J.G. ***Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters.*** Hingham, USA : s.n., 1985. Vol. 2. ISSN: 0920-5691.
46. González, Alejandro and Prieto, Flavio. ***Extracción de puntos característicos del rostro para medidas antropométricas.*** Colombia : s.n., 2010. Vol. 9. ISSN 1692-3324.
47. Guardado Medina, Ramón Osvaldo and Vallín González, Donato . ***Transformada de Fourier en aplicación en el diseño de filtros digitales para el procesamiento de imágenes.*** 2010.
48. Jadhao, Dattatray and Holambe, Raghunath . ***Feature Extraction and Dimensionality Reduction Using Radon and Fourier Transforms with Application to Face Recognition.*** Washington, USA : International Conference on Computational Intelligence and Multimedia Applications, 2007. ISBN:0-7695-3050-8.
49. Ripoll, M A. ***Reconocimiento y búsqueda selectiva de objetos en entornos no estructurados mediante apariencia multicanal y multiescala.*** 2006.

## **GLOSARIO DE TÉRMINOS**

**Algoritmo:** Conjunto ordenado y finito de operaciones que permite hallar la solución de un problema. En este trabajo se refiere al conjunto de procedimientos que se ejecutan de manera ordenada para lograr el reconocimiento automático de los individuos a partir de imágenes faciales.

**Eficacia:** Es la capacidad de alcanzar el efecto que se espera o se desea tras la realización de una acción.

**Eficiencia:** Es el uso racional de los medios para alcanzar un objetivo predeterminado.

**Eigenfaces:** Término en inglés por el cual se conoce al algoritmo de reconocimiento facial basado en el análisis de componentes principales (PCA).

**Método:** Modo de obrar o proceder. En el contexto del trabajo se refiere a la forma de ejecutar alguno de los pasos necesarios en el algoritmo de reconocimiento facial.

**Wavelets:** Una wavelet es una oscilación de onda, con una amplitud que comienza en cero, aumenta y luego disminuye de nuevo a cero.

**Eigenvalor:** los vectores propios, autovectores o eigenvectores de un operador lineal son los vectores no nulos que, cuando son transformados por el operador, dan lugar a un múltiplo escalar de sí mismos, con lo que no cambian su dirección.

**Píxel:** es la menor unidad homogénea que forma parte de una imagen digital, ya sea esta una fotografía, un fotograma de video o un gráfico.

**Fotograma:** a cada una de las imágenes individuales captadas por cámaras de video.

**Sinusoidal:** Curva que representa gráficamente la función trigonométrica seno.

**Iris:** es la membrana coloreada y circular del ojo que se separa de la cámara anterior de la cámara posterior. Posee una apertura de tamaño variable que comunica las dos cámaras: la pupila.

**Cámara anterior:** es una estructura del ojo que está limitada por delante por la córnea y por detrás por el iris. Este espacio se encuentra relleno de un líquido que se llama humor acuoso, el cual circula libremente y aporta los elementos necesarios para el metabolismo de las estructuras anteriores del ojo que no reciben aportes de la sangre como la córnea.

**Cámara posterior:** es un espacio situado detrás del iris, bañado por un fluido que se llama humor acuoso, el cual produce en esta zona los llamados procesos ciliares.

**Retina:** es un tejido sensible a la luz situado en la superficie interior del ojo. Es similar a una tecla donde se proyectan las imágenes.

**CENATAV:** Centro de Aplicaciones de Tecnologías de Avanzadas.

**RP:** Reconocimiento de Patrones.

**PCA:** Análisis de componentes Principales.

**LDA:** Análisis de Discriminante Lineal.

**DCT:** Transformada Discreta del Coseno.

**DWT:** Transformada Discreta de Wavelet.

**STFT:** Transformada de Fourier por intervalos

**RUP:** Proceso Unificado de Rational.

**XP:** Programación Extrema.

**UML:** Lenguaje Unificado de Modelado.

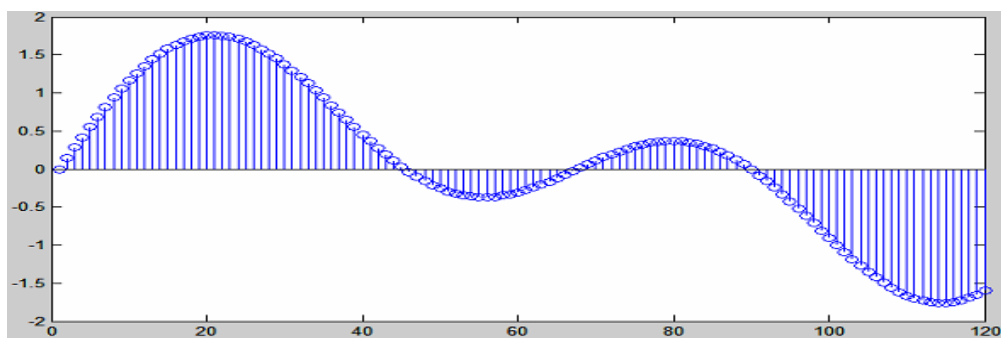
**IEEE:** Instituto de Ingenieros Eléctricos y Electrónicos.

## ANEXOS

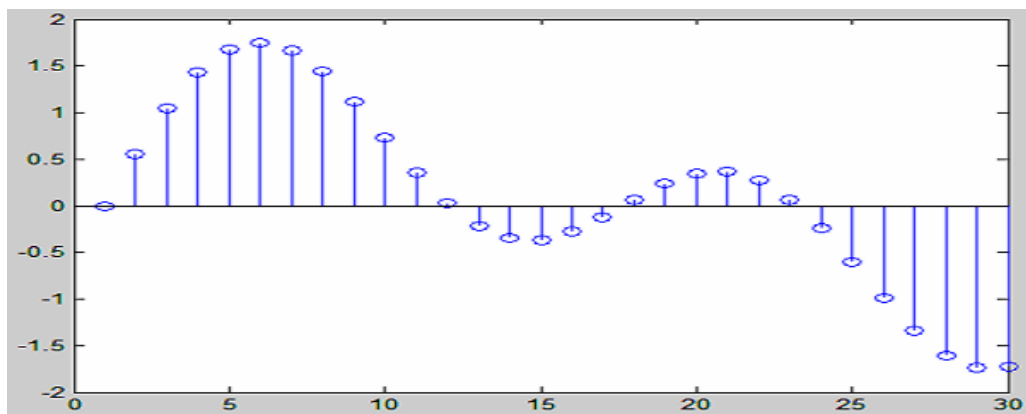
### Anexo 1

El proceso de decimación reduce la frecuencia de muestreo de la entrada  $F_x$  por un factor entero  $D$  y se realiza después de la etapa de filtrado.

Gráficamente lo que hace la decimación es tomar cada  $D$  muestras equidistantes de la señal original, y descarta las demás; dejando así una señal equivalente pero con diferente tasa de muestreo, como se puede ver en la Figura 19.



19 a. Señal original.



19 b. Señal decimada por un factor  $D = 4$ .

Figura 19. Decimación en tiempo.

## **Anexo2**

La base de datos de ORL (35) cuenta con un conjunto de imágenes faciales tomadas entre Abril de 1992 y Abril de 1994. La base de datos fue elaborada por AT&T *Laboratories Cambridge* en colaboración con el grupo de habla, visión y robótica del departamento de ingeniería de la Universidad de Cambridge.

Las imágenes de cada individuo fueron tomadas en diferentes sesiones. Las tomas incluyen variaciones de iluminación, expresión facial y el uso de anteojos. Todas las imágenes fueron tomadas contra un fondo homogéneo oscuro con el individuo en posición frontal.

Los archivos se encuentran en formato PGM, con una resolución de 92x112 píxeles con 256 niveles de gris por píxel.

La base de datos ORL (35) se puede descargar de forma gratuita de la página de internet de AT&T *Laboratories Cambridge* (35) (archivo comprimido, disponible en formatos tar y ZIP).

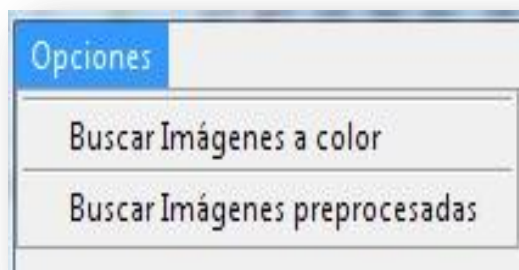
## **Anexo 3**

La escala de grises es empleada en las imágenes digitales en las que el valor de cada píxel posee un valor equivalente a una graduación de gris. Estas imágenes están representadas por sombras de grises, que van desde el negro más oscuro variando gradualmente en intensidad de grises hasta llegar al blanco.

Cuando se convierte una foto en escala de grises se puede representar un conjunto de colores en un tono de gris, o incluso poner cada color en una intensidad. Las escalas de grises son diferentes de las fotografías en blanco y negro, en las que los colores se codifican en blanco o en negro; la escala de grises ofrece una gama de tonalidades de gris entre ambos.

Las imágenes en escala de grises, emplean 8 bits para representar cada píxel lo que sólo permite una escala con 256 intensidades (o escalas de gris); es decir, 2 valores posibles para cada bit (0 y 1) elevado a 8 bits que se emplean para representar cada píxel, nos da 256 tonos de color diferentes que pueden representarse en una imagen en escala de grises.

**Anexo 4.** El usuario debe seleccionar el tipo de imágenes que serán procesadas para extraer el vector característico.



**Anexo 5.** El usuario debe indicar la ruta o dirección de la base de datos de entrenamiento y la imagen de prueba.



**Anexo 6.** En caso que las imágenes sean en color, estas se deben llevar a escala de grises.

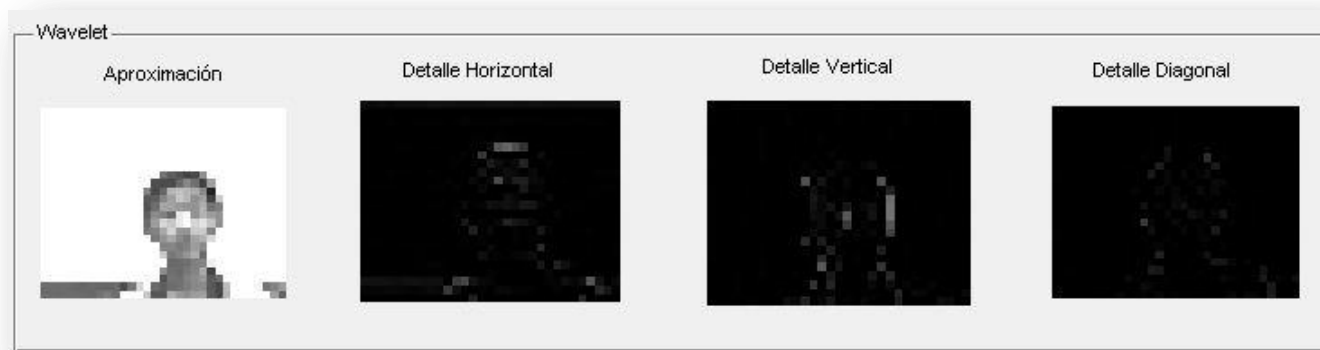


**Anexo 7.** El usuario debe seleccionar el tipo wavelet y el nivel de descomposición a utilizar para extraer el vector característico.

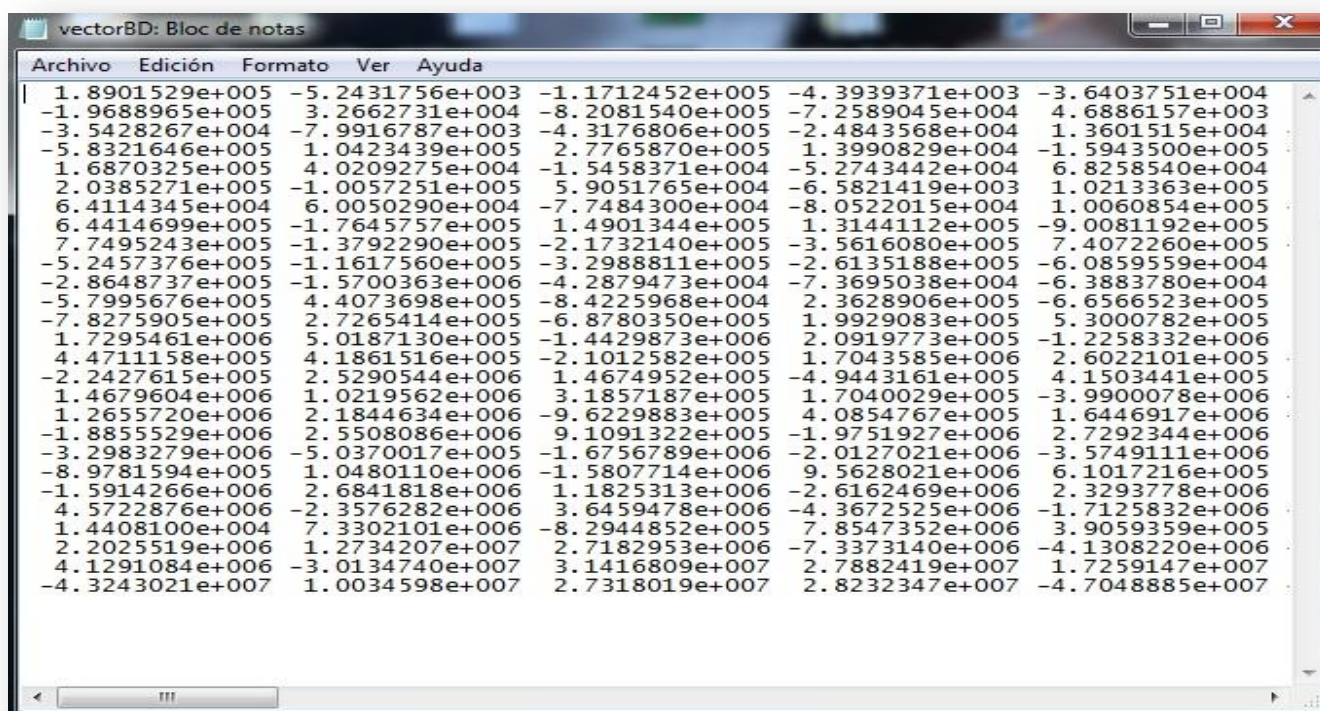


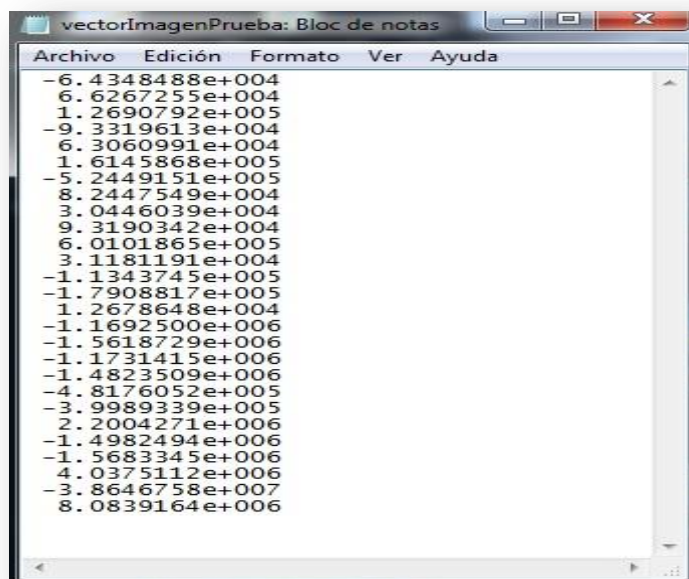


**Anexo 8.** El resultado de wavelet son 4 vectores o imágenes, uno de aproximación y 3 de detalles para cada imagen.



**Anexo 9.** Los vectores característicos pueden ser guardados para su posterior análisis en ficheros .xml que pueden abrirse con el bloc de notas para una mejor visualización del contenido. Cada columna será el vector de una imagen.





**Anexo 10.** Al presionar el botón Buscar Imagen más cercana se calcula la distancia euclidiana de cada vector de la base de datos a la imagen de pruebas. La imagen de la base de datos con menor distancia será la más semejante a la imagen que se desea reconocer o buscar a ver si existe en la base de datos.

