

FACEMATH: PROTOTIPO DE UN SISTEMA DE CONTROL DE ACCESO A  
TRAVES DEL ALGORITMO EIGENFACES

ANDRES FERNANDO BENAVIDES ARIAS

UNIVERSIDAD DE NARIÑO  
FACULTAD DE INGENIERIA  
PROGRAMA DE INGENIERIA DE SISTEMAS  
SAN JUAN DE PASTO  
2009

FACEMATH: PROTOTIPO DE UN SISTEMA DE CONTROL DE ACCESO A  
TRAVES DEL ALGORITMO EIGENFACES

ANDRES FERNANDO BENAVIDES ARIAS

Trabajo de Grado presentado como requisito parcial para optar al título de  
Ingeniero de Sistemas

CAMILO ARTURO LAGOS

Ing. Esp.  
Director del Proyecto

UNIVERSIDAD DE NARIÑO  
FACULTAD DE INGENIERIA  
PROGRAMA DE INGENIERIA DE SISTEMAS  
SAN JUAN DE PASTO  
2009

Las ideas y conclusiones aportadas en la tesis de grado son responsabilidad exclusiva de los autores. Artículo 1 del acuerdo N° 324 de octubre 12 de 1966, emanado del honorable Consejo Directivo de la Universidad de Nariño.

Nota de aceptación:

---

---

---

---

---

Firma del presidente del jurado

---

Firma del jurado

---

Firma del jurado

---

San Juan de Pasto, Octubre 9 de 2009

## RESUMEN

ESTE TRABAJO PRESENTA EL ANALISIS, DISEÑO E IMPLEMENTACION DE "FACEMATH: PROTOTIPO DE UN SISTEMA DE CONTROL DE ACCESO A TRAVÉS DEL ALGORITMO EIGENFACES". Y PRUEBAS DE FUNCIONALIDAD CON UNA BASE DE DATOS ESTANDAR Y REAL.

FACEMATH ESTA BASADO EN UNA ARQUITECTURA CLIENTE-SERVIDOR.

EN EL LADO DEL SERVIDOR SE ENCUENTRAN LOS MODULOS DE CONFIGURACION GENERAL DEL SISTEMA, EN EL CUAL, SE ESTABLECEN PARAMETROS COMO EL UMBRAL DE ERROR. EL REGISTRO DE USUARIOS, EN DONDE SE TOMAN LOS DATOS DE LOS USUARIOS Y FOTOGRAFIA DEL ROSTRO Y LA ADMINISTRACION DEL SERVICIO EN EL CUAL SE TIENE EL CONTROL DE ACCESO.

EN EL LADO DEL CLIENTE SE ENCUENTRA EL MODULO LECTOR DE ROSTROS, EN DONDE LOS USUARIOS HACEN PETICION PARA INGRESAR AL LUGAR RESTRINGIDO.

## **ABSTRACT**

THIS WORK PRESENTS THE ANALYSIS, DESIGN AND DEVELOPMENT OF "FACEMATH: PROTOTYPE OF AN ACCESS CONTROL SYSTEM THROUGH THE EIGENFACES ALGORITHM". AND FUNCTIONALITY TEST WITH A STANDARD AND REAL DATABASE.

FACEMATH IS BASED ON CLIENT-SERVER ARCHITECTURE.

ON SERVER SIDE ARE THE GENERAL CONFIGURATION MODULES WHICH PARAMETERS LIKE ERROR THRESHOLD IS SET UP, USER REGISTRATION MODULE WHERE USER INFORMATION AND FACE PHOTOGRAPH ARE CAPTURED. AND SERVICE MANAGEMENT WHICH HAS THE ACCESS CONTROL.

ON CLIENT SIDE IS FOUND THE FACE READER, WHERE USERS MAKE PETITION FOR ACCESS TO RESTRICTED PLACE.

## TABLA DE CONTENIDO

pág.

<b>1. INTRODUCCION .....</b>	<b>15</b>
<b>2. MARCO TEORICO.....</b>	<b>20</b>
<b>2.1. SISTEMAS DE CONTROL DE ACCESO.....</b>	<b>20</b>
2.1.1. Historia .....	20
2.1.2. Generales .....	21
2.1.3. De proximidad.....	21
2.1.4. Teclado.....	21
2.1.5. Biométricos.....	22
<b>2.2. ESTADO DEL ARTE EN RECONOCIMIENTO DE ROSTROS .....</b>	<b>27</b>
2.2.1. Enfoque psicológico y de neurociencias.....	27
2.2.2. Enfoque computacional.....	28
<b>2.3. CONCEPTOS DE ESTADISTICA .....</b>	<b>40</b>
2.3.1. Media .....	40
2.3.2. Varianza.....	41
2.3.3. Covarianza .....	42
2.3.4. Interpretación geométrica de varianza y covarianza .....	42
2.3.5. Propiedades de la matriz de covarianza. ....	44
<b>2.4. CONCEPTOS DE ALGEBRA LINEAL.....</b>	<b>46</b>
2.4.1. Matriz diagonal .....	46
2.4.2. Matriz de Hessenberg.....	46
2.4.3. Matriz tridiagonal.....	47
2.4.4. Base de un espacio.....	47
2.4.5. Cambio de base.....	47
2.4.6. Vectores y valores propios. ....	48
2.4.7. Diagonalización de matrices reales simétricas .....	50
2.4.7.1. Rotación de Jacobi.....	50
2.4.7.2. Rotación de Householder .....	53
<b>2.5. ANÁLISIS DE COMPONENTES PRINCIPALES .....</b>	<b>59</b>
2.5.1. Ejemplo inicial.....	59
2.5.2. Modelo matemático .....	63
<b>2.6. CONCEPTOS DE IMAGENES DIGITALES .....</b>	<b>67</b>
2.6.1. Pixel.....	67

2.6.2. Imagen digital.....	67
2.6.3. Escala de grises.....	67
2.6.4. Imagen como un vector.....	68
2.6.5. Filtros.....	69
2.6.6. Segmentación de imágenes.....	69
<b>2.7. ALGORITMO EIGENFACES.....</b>	<b>71</b>
2.7.1. Modelo matemático.....	71
2.7.2. Proceso de reconocimiento.....	73
2.7.3. Resumen gráfico del algoritmo.....	73
<b>3. METODOLOGIA Y HERRAMIENTAS PARA EL DESARROLLO DE FACEMATH.....</b>	<b>78</b>
<b>3.1. RUP: PROCESO RACIONAL UNIFICADO.....</b>	<b>78</b>
<b>3.2. .NET FRAMEWORK.....</b>	<b>79</b>
3.2.1. Información general y conceptual sobre .NET Framework.....	79
<b>3.3. WPF WINDOWS PRESENTATION FOUNDATION.....</b>	<b>80</b>
3.3.1. Introducción.....	80
3.3.2. XAML.....	80
3.3.3. Código subyacente.....	81
<b>3.4. WCF WINDOWS COMMUNICATION FOUNDATION.....</b>	<b>83</b>
3.4.1. Introducción.....	83
3.4.2. Mensajería y extremos.....	83
3.4.3. Protocolos de comunicaciones.....	84
<b>3.5. MICROSOFT VISUAL ESTUDIO.....</b>	<b>84</b>
3.5.1 Visual Studio 2008.....	85
<b>3.6. EXPRESSION STUDIO.....</b>	<b>86</b>
3.6.1. Expression Studio 2.....	86
3.6.1.1 Expression Blend 2.....	86
<b>4. ANALISIS, DISEÑO E IMPLEMENTACION DE FACEMATH.....</b>	<b>88</b>
<b>4.1. ANALISIS DE FACEMATH.....</b>	<b>88</b>
4.1.1. Funciones.....	88
4.1.2. Diagramas de Casos de Uso.....	90
<b>4.2 DISEÑO.....</b>	<b>92</b>
4.2.1. Diagramas de Paquetes.....	92
4.2.2. Diagrama de Clases.....	93



4.2.3. Diagrama de Secuencia .....	95
<b>4.3. IMPLEMENTACION .....</b>	<b>100</b>
<b>5. PRUEBAS Y RESULTADOS .....</b>	<b>105</b>
<b>5.1. MANTENER EL UMBRAL, VARIAR EL NÚMERO DE USUARIOS.....</b>	<b>106</b>
5.1.1. Prueba 1.1: umbral 0.2, 10 usuarios .....	106
5.1.2. Prueba 1.2: umbral 0.2, 15 usuarios .....	107
5.1.3. Prueba 1.3: umbral 0.2, 20 usuarios.....	107
5.1.4. Análisis de resultados prueba 1. ....	108
<b>5.2. VARIAR EL UMBRAL, MANTENER EL NÚMERO DE USUARIOS.....</b>	<b>109</b>
5.2.1 prueba 2.1 umbral 0.25, 15 usuarios.....	109
5.2.2 Prueba 2.2: umbral 0.30, 15 usuarios .....	109
5.2.3. Análisis de resultados prueba 2. ....	110
<b>5.3. PRUEBA CON BASE DE DATOS REAL.....</b>	<b>111</b>
<b>CONCLUSIONES .....</b>	<b>112</b>
<b>RECOMENDACIONES.....</b>	<b>113</b>
<b>REFERENCIAS BIBLIOGRAFICAS .....</b>	<b>114</b>

## ÍNDICE DE FIGURAS

	pág.
FIGURA 1: LLAVE DEL ANTIGUO EGIPTO. ....	20
FIGURA 2: LLAVE DE TAMBOR MODERNA. ....	21
FIGURA 3: CONTROL DE ACCESO DE PROXIMIDAD. ....	21
FIGURA 4: CONTROL DE ACCESO POR TECLADO. ....	22
FIGURA 5: SISTEMA DE IDENTIFICACIÓN DE ALPHONSE BERTILLION. ....	23
FIGURA 6: FINGER PRINTS, LIBRO DE SIR FRANCIS GALTON. ....	23
FIGURA 7: CONTROL DE ACCESO DACTILAR. ....	25
FIGURA 8: CONTROL DE ACCESO POR IRIS. ....	25
FIGURA 9: CONTROL DE ACCESO POR VOZ. ....	26
FIGURA 10: CONTROL DE ACCESO POR ROSTRO. ....	26
FIGURA 11: SISTEMA PROPUESTO POR CRAW, TOCK Y BENNET (1992)...	28
FIGURA 12: SISTEMA PROPUESTO POR GOVINDARAJU Y OTROS (1990). ...	29
FIGURA 13: PROPUESTA DE REISFELD Y YESHURUN (1992). ....	31
FIGURA 14: SISTEMA PROPUESTO POR YUILE, COHEN Y HALLINAN (1989), PARA EL OJO. ....	32
FIGURA 15: SISTEMA PROPUESTO POR YUILE, COHEN Y HALLINAN (1989), PARA LA BOCA. ....	32
FIGURA 16: SISTEMA PROPUESTO POR NIXON (1985). ....	33
FIGURA 17: SISTEMA PROPUESTO POR MANJUNATH Y OTROS (1992). ....	33
FIGURA 18: SISTEMA PROPUESTO POR KANADE (1977). ....	35
FIGURA 19: SISTEMA PROPUESTO POR TURK Y PENTLAND (1991). ....	36
FIGURA 20: SISTEMA PROPUESTO POR BRUNELLI Y POGGIO (1993). ....	36
FIGURA 21: SISTEMA PROPUESTO POR BRUNELLI Y POGGIO (1994). ....	37
FIGURA 22: SISTEMA PROPUESTO POR LUCAS (1995). ....	37
FIGURA 23: SISTEMA PROPUESTO POR LIU Y WECHSLER (1998). ....	39
FIGURA 24: INTERPRETACIÓN GEOMÉTRICA DE LA MEDIA EN UNA VARIABLE. ....	40
FIGURA 25: INTERPRETACIÓN GEOMÉTRICA DE LA MEDIA EN DOS VARIABLES. ....	40
FIGURA 26: INTERPRETACIÓN GEOMÉTRICA DE DATOS CENTRADOS. ....	41
FIGURA 27: INTERPRETACIÓN DE LA VARIANZA COMO EL NIVEL DE SEÑAL. ....	41
FIGURA 28: INTERPRETACIÓN GEOMÉTRICA DE R COMO EL COSENO DEL ÁNGULO QUE FORMAN LOS VECTORES DE LAS DESVIACIONES CON RESPECTO A SUS RESPECTIVAS MEDIAS DE X Y DE Y. ....	43
FIGURA 29: DEPENDENCIA LINEAL DE DOS VARIABLES. ....	44
FIGURA 30: MATRIZ DE HESSENER INFERIOR. ....	46
FIGURA 31: MATRIZ DE HESSENER SUPERIOR. ....	46
FIGURA 32: MATRIZ TRIDIAGONAL. ....	47

FIGURA 33: INTERPRETACIÓN GEOMÉTRICA DE U, COMO UNA ROTACIÓN DE LOS EJES P1,P2 EN P'1,P'2.....	48
FIGURA 34: EFECTO DEL OPERADOR DE HOUSEHOLDER SOBRE EL VECTOR X .....	55
FIGURA 35: EJEMPLO DE UNA MASA SUJETA A UN RESORTE.....	60
FIGURA 36: VISTA DE LOS DATOS EN 3 DIMENSIONES.....	61
FIGURA 37: ROTACIÓN DE LA VISTA DE LOS DATOS. ....	62
FIGURA 38: NUEVO SISTEMA DE REFERENCIA PARA LOS DATOS DEL RESORTE. ....	63
FIGURA 39: IMAGEN DIGITAL. ....	67
FIGURA 40: IMAGEN EN ESCALA DE GRISES.....	68
FIGURA 41: IMAGEN BINARIA .....	68
FIGURA 42: IMAGEN COMO UN VECTOR. ....	68
FIGURA 43: PROCESO DE FILTRADO DE UNA IMAGEN. ....	69
FIGURA 44: PROCESO DE SEGMENTACIÓN DE IMÁGEN.....	70
FIGURA 45: EJEMPLO IMAGENES DE ENTRENAMIENTO.....	74
FIGURA 46: REPRESENTACIÓN GRAFICA DE LOS EIGENFACES. ....	76
FIGURA 47: IMAGEN COMO UNA COMBINACIÓN DE PESOS EN FACESPACE.....	76
FIGURA 48: IMAGEN DE PRUEBA PARA SER RECONOCIDA. ....	76
FIGURA 49: INTERFAZ GRAFICA DEFINIDA MEDIANTE XAML. ....	81
FIGURA 50: RESULTADO DE LA PROGRAMACIÓN DEL CÓDIGO SUBYACENTE DE UN BOTÓN.....	82
FIGURA 51: DIAGRAMA DE CASO DE USO GENERAL.....	90
FIGURA 52: DIAGRAMA DE CASO DE USO, MODULO CONFIGURACIÓN.....	90
FIGURA 53: DIAGRAMA DE CASO DE USO, MODULO ADMINISTRACIÓN USUARIOS.....	90
FIGURA 54: DIAGRAMA DE CASO DE USO, MODULO ADMINISTRACIÓN SERVICIO.....	91
FIGURA 55: DIAGRAMA DE PAQUETES PRINCIPAL. ....	92
FIGURA 56: DIAGRAMA DE PAQUETE SERVER.....	92
FIGURA 57: PAQUETE MATH. ....	93
FIGURA 58: PAQUETE ADMINISTRACIÓN SERVICIO. ....	93
FIGURA 59: PAQUETE ADMINISTRACIÓN USUARIOS.....	94
FIGURA 60: PAQUETE CONFIGURACIÓN. ....	94
FIGURA 61: PAQUETE CLIENTE. ....	95
FIGURA 62: RUN CLASE CONFIGURATION.....	96
FIGURA 63: RUN CLASE DATABASE.....	97
FIGURA 64: RUN CLASE SERVICE. ....	98
FIGURA 65: RUN CLASE CLIENT .....	99
FIGURA 66: INTERFAZ GRAFICA DEL SERVIDOR. ....	100
FIGURA 67: INTERFAZ GRAFICA MODULO CONFIGURACIÓN.....	101

FIGURA 68: INTERFAZ GRAFICA MODULO ADMINISTRACIÓN DE USUARIOS.....	102
FIGURA 69: INTERFAZ GRAFICA MODULO ADMINISTRACIÓN DEL SERVICIO.....	103
FIGURA 70: INTERFAZ GRAFICA MODULO CLIENTE.....	104
FIGURA 71: RESULTADO DEL RECONOCIMIENTO COMO VERDADERO-POSITIVO.....	105
FIGURA 72: RESULTADO DEL RECONOCIMIENTO COMO FALSO-POSITIVO. 105	105
FIGURA 73: RESULTADO DEL RECONOCIMIENTO COMO VERDADERO-NEGATIVO.....	106
FIGURA 74: RESULTADO DEL RECONOCIMIENTO COMO FALSO-NEGATIVO.....	106

## INDICE DE TABLAS

	pág.
TABLA 1: EJEMPLO DE DATOS DE DOS VARIABLES.....	40
TABLA 2: EJEMPLO DE LOS DATOS TOMADOS POR LA CÁMARA.....	61
TABLA 3: TABLA DE FUNCIONES DE FACEMATH.....	88
TABLA 4: DISEÑO DE LA PRUEBA 1.1.....	107
TABLA 5: DISEÑO DE LA PRUEBA 1.2.....	107
TABLA 6: DISEÑO DE LA PRUEBA 1.3.....	108
TABLA 7: RESULTADOS DE LA PRIMERA 1.....	108
TABLA 8: DISEÑO PRUEBA 2.1.....	109
TABLA 9: DISEÑO DE LA PRUEBA 2.2.....	110
TABLA 10: RESULTADOS PRUEBA 2.....	110

## INDICE DE ANEXOS

	<b>pág.</b>
ANEXO 1. BASE DE ROSTROS AT & T.....	121
ANEXO 2. BASE DE ROSTROS REAL.....	131

## 1. INTRODUCCION

La habilidad que tiene el ser humano para reconocer un rostro es realmente sorprendente, una persona puede memorizar un gran número de rostros y reconocerlos así estos presenten rotaciones o cambien con el tiempo. Aunque parezca algo tan natural para las personas, lograr que una computadora reconozca un rostro, como para por ejemplo, permitirle o no el acceso a un lugar restringido, es aun una tarea muy compleja, problema que ha incentivado nuevas líneas de investigación y aplicaciones tan diversas que van desde la seguridad hasta interfaz hombre – máquina.

Un sistema de control de acceso biométrico tiene que realizar dos grandes actividades: la primera es el de registro, en el cual, las personas son ingresadas al sistema. El proceso de registro consiste en que una persona se muestra ante el sistema indicando su identidad y característica que lo identifica, como por ejemplo: la huella digital, una tarjeta electrónica o en este caso, el rostro.

El sistema almacena esta información y crea un modelo electrónico para uso en el siguiente paso.

El segundo paso es el de reconocimiento. La persona no presenta su identidad, sino la característica que lo identifica (rostro), el sistema lee esta característica y crea un modelo electrónico de la persona, el cual debe comparar con los modelos ya existentes en su base datos. Si al comparar el nuevo modelo con los existentes en la base datos, el error es menor a cierto umbral, el sistema le permite el acceso, sino rechaza a la persona.

El algoritmo utilizado para crear el modelo es EigenFaces.

En este documento se presenta el trabajo de grado para optar al título de Ingeniero de Sistemas, en el cual se presenta el desarrollo de “FaceMath; Prototipo de un Sistema de control de acceso a través del algoritmo EigenFaces”.

El resto de este documento está organizado de la siguiente manera. En la siguiente sección se especifica el tema de la propuesta, el cual es enmarcado dentro de una línea de investigación y delimitado, se describe el problema objeto de estudio, se describen los objetivos generales y específicos del proyecto. En la sección 2 se presenta el Marco Teórico, el cual incluye los diferentes sistemas de control de acceso y el estado general del arte en Reconocimiento de Rostros respectivamente, conceptos de Estadística y Algebra Lineal necesarios para la comprensión del algoritmo, el Análisis de Componentes principales y el algoritmo Eigenfaces. En la sección 3, se presentan la metodología y herramientas

utilizadas para el desarrollo del proyecto. En la sección 4, se encuentra el Análisis, Diseño e Implementación de FaceMath. En la sección 5, se presenta las pruebas realizadas y resultados. En la sección 6, se presentan las conclusiones de este trabajo y finalmente en la sección 7, se presenta las recomendaciones.

## **TEMA**

**Título.** FaceMath. Prototipo de un Sistema de Control de Acceso a través del algoritmo Eigenfaces

**Línea de Investigación.** El presente trabajo de grado, se encuentra inscrito bajo la línea de Software y manejo de Información. Se enmarca dentro del área de Inteligencia Artificial.

**Alcance y delimitación.** FaceMath es un prototipo que contempla las etapas de un sistema biométrico para el control de acceso a través de la imagen del rostro, es decir, la adquisición de fotografías para la base de entrenamiento y la identificación de personas también a base de una fotografía [1], la cual es tomada por una cámara.

Tanto en la etapa de entrenamiento y de reconocimiento está enfocada al algoritmo Eigenfaces [2].

Se desarrolló una interfaz Cámara-Computador para obtener imágenes del rostro y algoritmos de pre procesamiento para las imágenes obtenidas.

Se construyó un dispositivo electrónico para la simulación de la respuesta del sistema ante una persona identificada positivamente.

Se desarrolló una interfaz gráfica al usuario que le permite una interacción sencilla con el sistema.

Para el prototipo se realizaron pruebas de reconocimiento.

## **PROBLEMA OBJETO DE ESTUDIO**

**Descripción del problema.** Para las empresas y los gobiernos, la seguridad es un tema de vital importancia. Si no se cuentan con sistemas de seguridad, una persona ajena a la entidad, podrían ingresar a esta, y causar grandes pérdidas, ya sean materiales o de información, es por ello, que se protegen los lugares más importantes y vulnerables con sistemas de control de acceso. En la actualidad existen muchos sistemas de control de acceso a lugares restringidos, algunos se basan en “algo que tienes”, como por ejemplo, una credencial o una tarjeta electrónica. El problema de este tipo de sistemas es que una credencial puede ser



falsificada, y una tarjeta electrónica puede ser clonada o utilizada por otra persona sin más verificación.

Otro tipo de sistemas se basan en “algo que tu sabes”, como la contraseña o password, que puede ser obtenido o adivinado, o utilizado por diferentes personas. Los buenos password son difíciles de recordar.

Es por ello es que en los últimos años se han estudiado y desarrollado sistemas biométricos como métodos efectivos de seguridad, ya que no pueden ser perdidos, robados o adivinados [3].

De los diferentes tipos de sistemas biométricos, los que utilizan huellas dactilares, iris del ojo, etc., necesitan de sofisticados dispositivos electrónicos, resultando costosa su implementación, al mismo tiempo que resulta incomoda su utilización por parte de los usuarios finales [4].

En Colombia gran parte de la tecnología de la biometría utilizada es importada de países como Estados Unidos, Alemania y China, siendo la más común, el reconocimiento de huellas dactilares, pero no hay industria colombiana que produzca tecnología de reconocimiento de rostros.

Las investigaciones que se han llevado a cabo en Colombia acerca del reconocimiento de rostros se han enfocado en dos direcciones: de tipo psicológicas [5] y de carácter computacional, destacándose una investigación teórica en [6] y de implementación de sistemas que solo reconocen rostros. Pero no de aplicaciones a la seguridad, y más específicamente, al control de acceso, desconociéndose las ventajas de su utilización.

En el Departamento de Nariño, no se tiene antecedentes del desarrollo de un sistema biométrico para control de acceso, por lo cual se plantea del desarrollo de este prototipo para avanzar en este campo del conocimiento.

**Formulación del problema.** ¿Es factible el desarrollo de un prototipo de sistema de control de acceso basado en el algoritmo Eigenfaces para permitir o no la entrada de personas a un lugar restringido?

## **OBJETIVOS**

**Objetivo general.** Proponer un sistema de control de acceso a través de la implementación prototípica del Algoritmo EigenFaces.

**Objetivos específicos.**

- Recolectar, filtrar y analizar información relevante para el desarrollo del proyecto.
- Analizar, diseñar y desarrollar un sistema biométrico para el control de acceso.
- Analizar, diseñar y desarrollar una interfaz Camara-Computador.
- Analizar, diseñar y desarrollar módulos para preprocesamiento de imagen.
- Analizar e implementar el algoritmo Eigenfaces.
- Analizar, diseñar y desarrollar un dispositivo electrónico para simular la respuesta del sistema.
- Realizar pruebas y analizar los resultados.
- Dar a conocer los resultados y conclusiones.
- Desarrollar el prototipo de sistema de control de acceso.

## JUSTIFICACION

Para las empresas y los gobiernos, la seguridad es uno de los ejes centrales, invirtiendo grandes cantidades de dinero para sistemas cada vez más seguros. De acuerdo con la empresa Internacional Biometric Group (una de las empresas líderes en el desarrollo de sistemas biométricos) se consideró que las ventas de sistemas biométricos crecerían de \$600.8 millones de dólares en el año 2006 a \$4035 millones de dólares para el año 2009. [7]

El reconocimiento de rostros es un área de investigación que cada vez tiene un mayor auge entre la comunidad académica. Del estudio de las principales técnicas se puede lograr una mayor comprensión del tema y proponer nuevos y más eficientes algoritmos para dicha tarea.

De los sistemas biométricos, los que utilizan al rostro para reconocimiento son más sencillos y cómodos para los usuarios. Ya que pueden ser instalados en lugares desapercibidos, y cumplir su función sin mucha cooperación de los usuarios.

Además de las aplicaciones de seguridad se pueden citar algunas otras [8]:

- Videoconferencia. Tiene que ver con localizar la imagen del individuo en una secuencia de imágenes para poder hacer un seguimiento. También saber si está o no presente, y quien es.
- Interfaces hombre-máquina. Seguridad de *log in* en una computadora.
- Detección de expresiones faciales (se considera un subcampo del reconocimiento de rostros en general):

- Interfaces inteligentes, detectores de cansancio para conductores, aplicaciones médicas, etc.
- Base de datos multimedia. El incremento de imágenes en el World Wide Web sugiere cada vez más la existencia de aplicaciones CBIR de recuperación de imágenes. En sistemas con bibliotecas digitales de terabytes de video y audio la clasificación de las imágenes cumple un papel fundamental.

## 2. MARCO TEORICO

### 2.1. SISTEMAS DE CONTROL DE ACCESO

Un sistema de control de acceso se define como aquel que permite restringir el acceso a un área a personas no autorizadas.

#### 2.1.1. Historia. Desde hace más de 4000 años el ser humano ha utilizado las cerraduras con llave como sistema de control de acceso.

Según la tradición griega el inventor de la cerradura es Teodoro de Samos, en tiempos del poeta Homero, pero la verdad es que las primeras puertas con cerraduras aparecieron en Egipto hace 4,000 años. Estas puertas podían abrirse desde fuera con llaves y candados hechos de madera.

En un principio esas puertas se cerraban mediante pestillos de madera que se introducían lateralmente en el marco o en una grapa fijada en la propia hoja de la puerta; después se aseguraron con varios pernos, que se elevaban fácilmente y penetraban en los orificios practicados en el pestillo gracias a la acción de las llaves. (Ver figura 1).



Figura 1: Llave del antiguo Egipto.

La evidencia más antigua de un candado fue encontrado en la ciudad de Ninive, Mesopotamia, y funcionaba con ese principio básico. Los modelos de candados egipcios eran empleados solamente por personas ricas, pero fueron los griegos quienes perfeccionaron llaves y candados disponibles para uso popular.

Ya con posterioridad, los romanos idearon candados que podían abrirse con llaves que cargaban como si fueran anillos.

El Inventor Linus Yale, patentó una cerradura de tambor de pines en 1848, un diseño que ha sido posteriormente modificado por su hijo, Linus Yale Junior., en

1861. El diseño de Yale Junior es muy similar a las cerraduras de tambor de pines que se hacen hoy en día [9]. (Ver figura 2).

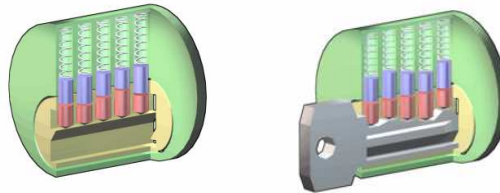


Figura 2: Llave de tambor moderna.

### 2.1.2. Generales

- Guardias y vigilantes
- Puertas con trancas de madera o metal
- Candados con llaves de metal
- Cerraduras o cerrojos mecánicos

**2.1.3. De proximidad.** Ejemplos de este tipo utilizan dispositivos basados en tarjetas de proximidad, código de barras o banda magnética. En este sistema, las tarjetas tienen información del usuario, cuando el usuario acerca la tarjeta al lector, el sistema registra la información y le permite el acceso. (Ver figura 3).

Características:

- Las tarjetas son fáciles de clonar.
- Bajo costo.
- Requieren de mantenimiento.
- La vida útil de las tarjetas es corta.



Figura 3: Control de Acceso de proximidad.

**2.1.4. Teclado.** Permite accionar cualquier cerradura eléctrica, portones automáticos, barreras de control de acceso de vehículos y muchos otros. En este sistema, el usuario tiene que digitar una clave, la cual es verificada por el sistema,

si dicha clave existe, la asocia al usuario al que pertenece, registrando la información y permitiéndole el acceso. (Ver figura 4).

Características:

- Bajo nivel de seguridad
- Programación en la misma unidad
- Fácil de obtener códigos



Figura 4: Control de acceso por teclado.

**2.1.5. Biométricos.** Son los sistemas de control de acceso moderno.

El objetivo de la biometría consiste en la elaboración de métodos automatizados para la identificación de personas a través de características físicas como las huellas dactilares, los ojos, el rostro, etc. Características de comportamiento como la voz, la manera de caminar o la firma. Esta tecnología se basa en la premisa de que cada persona posee estas características únicas, y por lo tanto, comparables e identificables.

**2.1.5.1. Historia de la biometría.** Históricamente los primeros en utilizar medidas físicas para identificación, fueron los chinos en el siglo XIV, donde se utilizaban ciertas impresiones de los dedos.

En el año de 1877, Alphonse Bertillion desarrolló y utilizó de manera sistemática un método para identificar criminales, en base a diferentes medidas del cuerpo y la cabeza. (Ver figura 5).



Figura 5: Sistema de identificación de Alphonse Bertillion.  
Imagen del sistema de Bertillion en [http://es.wikipedia.org/wiki/Alphonse\\_Bertillon](http://es.wikipedia.org/wiki/Alphonse_Bertillon)

En el año de 1892, Sir Francis Galton, antropólogo británico y primo de Charles Darwin, publicó sus investigaciones de clasificación de huellas dactilares en su libro *Finger Prints*. (Ver figura 6).

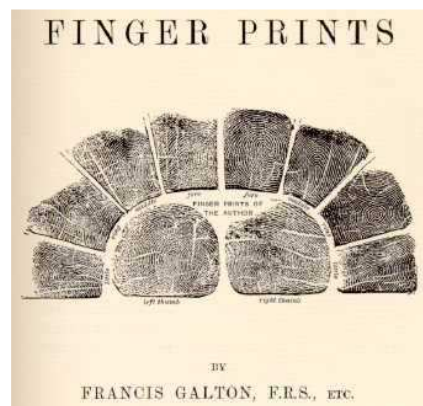


Figura 6: *Finger Prints*, libro de sir Francis Galton.  
Imagen disponible en <http://etext.lib.virginia.edu/railton/wilson/galtonfp.html>

Utilizando las investigaciones de Galton, Edward Richard Henry, desarrolló entre 1896 y 1897 lo que está considerado como el primer sistema policiaco para identificación a través de huellas dactilares.

Para un desarrollo más profundo véase [10].

**2.1.5.2. Descripción de un sistema biométrico.** Un sistema biométrico consiste en dos grandes pasos, y son [11]:

El primero, consiste en que una persona debe registrarse en sistema. Durante el proceso de registro, el sistema captura el rasgo característico de la persona, como por ejemplo la huella digital o la imagen del rostro, y lo procesa para crear una representación electrónica denominada “modelo de referencia” o “modelo de entrenamiento”.

El modelo de entrenamiento debe ser guardado en una base de datos o en algún otro lugar del cual será extraído cuando se lo necesite.

El segundo paso, depende si la función del sistema biométrico es la verificación de la identidad de la persona o reconocer a la persona. En el caso de verificación, la persona informa al sistema cual es su identidad ya sea presentando una tarjeta de identificación, dando una clave o proporcionando su huella digital o imagen del rostro. El sistema captura el rasgo característico de la persona y lo procesa para crear también una representación electrónica llamada “modelo en vivo”. Por último, el sistema compara el modelo en vivo con el modelo de referencia de la persona. Si ambos modelos son idénticos o la diferencia es menor que un determinado umbral, se considera una verificación adecuada.

Cuando la función del sistema es la de reconocimiento, la persona no le informa al sistema cual es su identidad. El sistema tan solo captura el rasgo característico de la persona y lo procesa para crear el modelo en vivo. Luego el sistema procede a comparar los modelos respectivos para determinar la identidad de la persona en la base de datos.

**2.1.5.3. Dactilar.** Generalmente el más utilizado, desde el log on en una computadora, hasta control de nómina. (Ver figura 7).

A lo ya mencionado se puede adicionar que en 1963 el investigador Trauring del Laboratorio Hughes, publica un documento donde describe un proceso para reconocimiento automático de huellas dactilares.

En 1975, NIST se enfocó en digitalizar la información de las huellas dactilares; estudio los efectos de la compresión de imágenes en el proceso de clasificación, extracción de características y emparejado de las mismas. El resultado de estas investigaciones fue el desarrollo del algoritmo M40, el primer algoritmo operacional utilizado por el FBI [12].





Figura 7: Control de acceso dactilar.

**2.1.5.4. Iris.** Utiliza el iris del ojo para reconocimiento. (Ver figura 8).

El oftalmólogo Frank Burch propuso en 1936, el concepto de usar el iris como un método para reconocer a un individuo.

En 1985, los oftalmólogos Leonard Flom y Aran Safir propusieron el concepto de que dos iris de diferentes personas no pueden ser iguales, y enfocaron a Jhon Daugman a desarrollar un algoritmo para la identificación automática del iris humano.

En 1993, la Defense Nuclear Agency, comenzó a trabajar con IriScan Inc, para probar y liberar un prototipo de sistema de reconocimiento del iris, que finalmente fue lanzado comercialmente en 1995 [13].



Figura 8: Control de acceso por iris.

**2.1.5.5. Voz.** Algunas de las cualidades más importantes de la voz son su naturalidad, versatilidad y escasa intrusividad, que hacen de ella un método biométrico agradable para el usuario y adecuado para un buen número de aplicaciones. (Ver figura 9).

En 1960, un profesor sueco, Gunnar Fant publicó un modelo describiendo los componentes fisiológicos de la producción del sonido en el habla. Sus descubrimientos fueron basados en el análisis de rayos X, hechos por individuos que hacían sonidos específicos. Este estudio fue ampliado por Joseph Perkel, que adicionó al modelo un estudio de la fisiología de la lengua y la mandíbula.

En 1976, Texas Instruments desarrolló un prototipo de sistema de reconocimiento de voz que fue probado por la fuerza aérea estadounidense [14]



Figura 9: Control de acceso por voz.

**2.1.5.6. Rostro.** Es uno de los de mayor crecimiento en cuanto a inversión y a expectativas ha tenido, ya que es muy natural utilizar el rostro para reconocer a una persona, además, su implementación es relativamente sencilla y económica [4]. (Ver figura 10).

De su desarrollo se hablará con más detalle en las siguientes secciones.



Figura 10: Control de acceso por rostro.

## **2.2. ESTADO DEL ARTE EN RECONOCIMIENTO DE ROSTROS**

El reconocimiento de rostros por medios computacionales forma parte del área llamada biometría (medida de la vida).

Existen dos grandes enfoques para el estudio de reconocimiento de rostros:

- El psicológico y de neurociencias
- El reconocimiento por medio de computadoras.

Algunos de los resultados de las investigaciones del reconocimiento de rostros desde el punto de vista psicológico y neurociencias se han utilizado para proponer diversos métodos algorítmicos para el reconocimiento automatizado.

Sin embargo, el enfoque computacional es el de mayor interés en este trabajo, por lo que, se mencionan algunos resultados del enfoque psicológico y de neurociencias; y se prestará mayor importancia al computacional.

### **2.2.1. Enfoque psicológico y de neurociencias**

Algunos de los resultados obtenidos por este enfoque, son [15]:

**En cuanto a la existencia o no, de un sistema de neuronas para el reconocimiento de rostros.**

- Las caras son más fáciles de recordar por los humanos que cualquier otro objeto, cuando estas se les presentan en rotación invertida [5].
- Los pacientes con *prosopagnosia* no pueden reconocer caras familiares. Aunque perciben los ojos, nariz, boca, pelo, etc., no pueden conjurar todos los elementos para propósitos de identificación. Distinguen si un objeto es una cara o no, pero no pueden identificarla.
- Los neonatos prefieren ver más a una cara que a cualquier otro objeto.

**En cuanto a que si el reconocimiento se realiza de forma holística o en características específicas.**

Se ha encontrado que ambas informaciones son esenciales para percepción y reconocimiento. Algunos estudios sugieren que la información holística es utilizada como un fondo para luego afinar la percepción por características.

**En cuanto a que características faciales son importantes para el reconocimiento de rostros.**

Para percibir y recordar rostros son importantes la forma de la cara, el pelo, los ojos y la boca. La nariz tiene un papel insignificante. La parte superior de la cara es la más importante para el reconocimiento que la parte inferior. Las caras más atractivas son más fáciles de recordar, seguida de las menos atractivas y finalmente las caras normales.

**2.2.2. Enfoque computacional.** A continuación se describen los principales métodos y técnicas que se han utilizado para realizar pre procesamiento computacional tendente a identificación de rostros.

**2.2.2.1. Segmentación de rostros.** Craw y otros [16] describen un método para extraer el área de la cabeza a partir de una imagen. Utilizaban una escala de imagen jerárquica y una plantilla de escala.

Craw, Tock Y Bennet [17] describen un sistema para reconocer y medir características faciales (ver figura 11). Ellos tratan de localizar 40 puntos característicos a partir de una imagen en escala de gris; dichos puntos se seleccionaron con base a la propuesta de Shepherd, el cual también fue utilizado como un criterio de juicio. El sistema utiliza una búsqueda jerárquica de grueso a fino. La plantilla se basa en el principio de transformación poligonal aleatoria propuesta por Grenander.

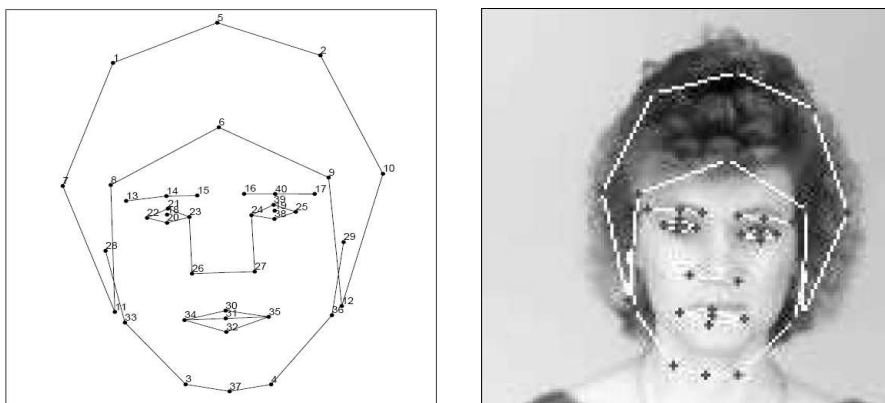


Figura 11: Sistema propuesto por Craw, Tock Y Bennet (1992).

La localización aproximada, escala y orientación de la cabeza se obtiene por una deformación iterativa de la plantilla total por medio de escalamiento aleatorio, traslación y rotación.

Govindaraju y otros [18] consideran un modelo para localizar la cara de una imagen desordenada, su técnica utiliza una plantilla deformable que es ligeramente diferente a la propuesta por Yuille y otros [19] (ver figura 12). Trabajan sobre los bordes de la imagen, basado en esto su plantilla sobre el contorno de la cabeza. La plantilla se compone de tres segmentos que se obtienen a partir de las discontinuidades de curvatura del contorno de la cabeza. Estos tres segmentos forman la línea del lado derecho, la línea del lado izquierdo y la línea del cabello. Cada una de estas curvas se le asigna una cuatri-tupla que consiste de la longitud de la curva, la cuerda en forma vectorial, el área encerrada entre la curva y la cuerda y el centroide de dicha área. El centro de dichos tres segmentos da la localización del centro de la cara y posteriormente, las plantillas permiten trasladar, escalar y rotar de acuerdo a ciertos modelos.

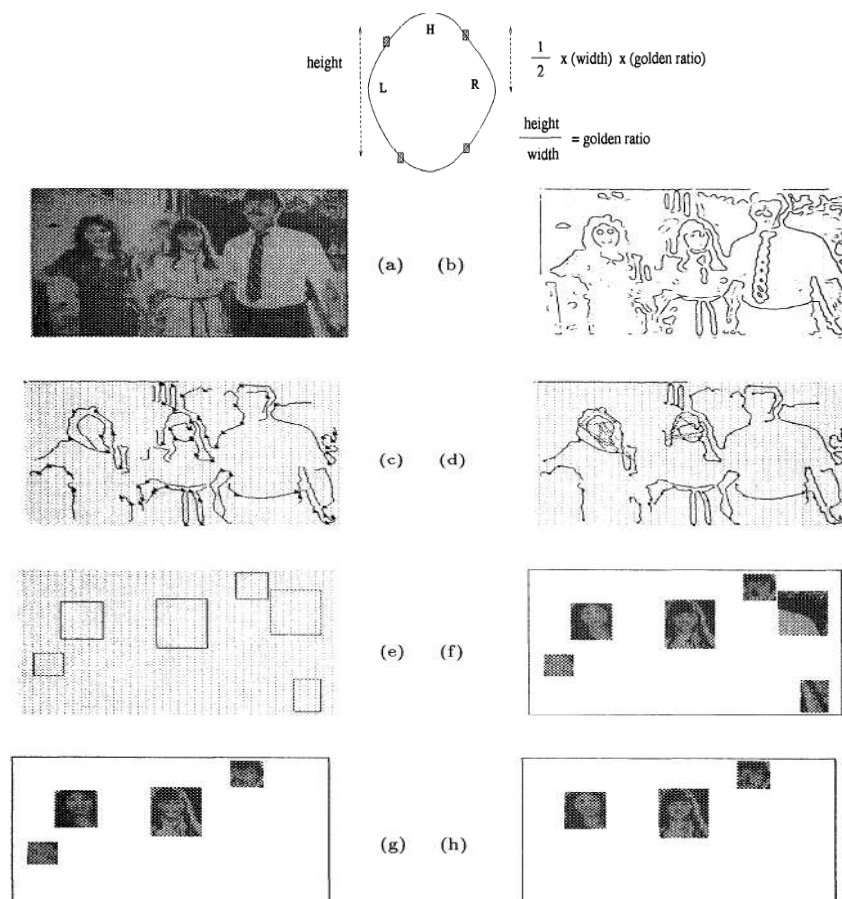


Figura 12: Sistema propuesto por Govindaraju y otros (1990).

En una propuesta de Brunelli y Poggio [20], la cara se segmenta a partir de un fondo moderadamente desordenado. El enfoque involucra trabajar con la

intensidad de la imagen de la cara así, como con los bordes encontrados mediante el método de Canny, las tareas de pre procesamiento incluyen localizar la intersección de puntos de bordes, asignar etiquetas a segmentos contiguos de bordes y enlazar segmentos contiguos en bordes a puntos de intersección.

La cara se aproxima mediante la utilización de una elipse como herramienta analítica. Pares de segmentos de bordes etiquetados como  $L_i$ ,  $L_j$  se ajustan a una ecuación lineal de la elipse.

El conjunto de parámetros resultante se verifica contra la tasa de aspectos de la cara, y si es satisfactoria, se incluye en la clase del conjunto de parámetros para la selección final.

Se reporta una precisión superior al 80% en el proceso cuando se aplicó a un conjunto de 48 imágenes.

**2.2.2.2. Extracción de características.** Los primeros trabajos de reconocimiento de rostros fueron realizados por Sakai y otros [21] en el año de 1969. Utilizaron una imagen digitalizada con 8 niveles de gris.

El trabajo se realizó sobre un conjunto de datos consistente de imágenes frontales de caras, y se utilizó una cuadrícula de 3x3 para determinar los pixeles que presentan el mayor valor del gradiente, de tal forma que la cantidad de información se reduce a elementos esenciales. Estos pixeles se conectan a pixeles vecinos que exhiban características similares para formar líneas y segmentos de contorno. Un enfoque de "grueso a fino" se utiliza para determinar características individuales de la cara. El recogimiento no diferencia entre caras diferentes, determina únicamente la existencia de una cara en la imagen. Los autores anotan que el procedimiento empleando presenta una dependencia en la dirección del iluminación, cambios en esta, causa problemas en el enfoque.

Reisfeld y Yeshurun [22], utilizan un operador de simetría generalizado con el propósito de encontrar los ojos y boca de una cara (ver figura 13). Su motivación proviene de la naturaleza simétrica de la cara sobre una línea vertical que pase a través de la nariz. Simetrías subsecuentes se permiten dentro de las características de ojos, nariz y boca. El operador de simetría localiza puntos en la imagen correspondiente valores altos de una medida de simetría. No es aparente, como los autores determinan si una característica es un ojo o nariz. Se indica que el procedimiento presenta una superioridad sobre otros esquemas basados en correlación, similares al de Baron [23] en el sentido de que dicho esquema es independiente de la escala u orientación. El método es computacionalmente intenso, se mencionan tasas de éxito de 95%, con la restricción de que la cara ocupe entre el 15-60% de la imagen.

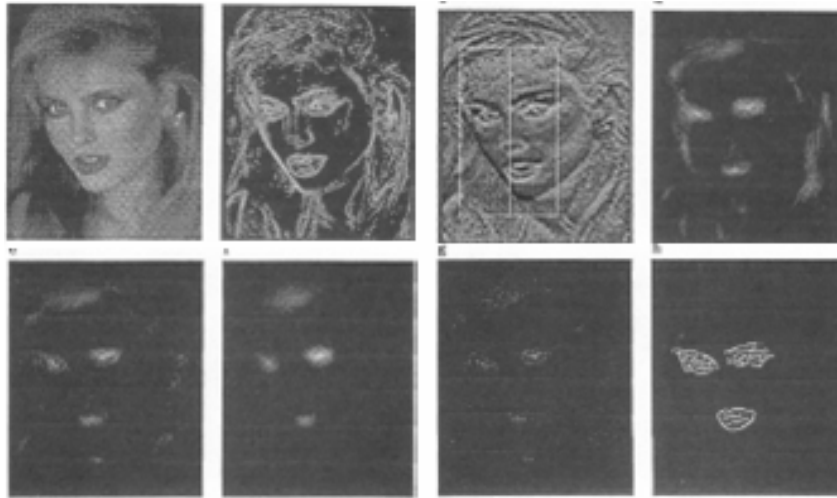


Figura 13: Propuesta de Reisfeld y Yeshurun (1992).

Yuile, Cohen y Hallinan [19], extraen características faciales utilizando plantillas deformables, las cuales permiten trasladar, rotar y deformar con el propósito de ajustarse a la mejor representación de la forma actual de la imagen. El pre procesamiento se realiza a la imagen de intensidad inicial para las representaciones de picos y valles. Filtros morfológicos se utilizan para determinar dichas representaciones.

Las plantillas para el ojo tienen 11 parámetros consistentes de los arcos superiores e inferiores, el círculo para el iris, los puntos centrales y el ángulo de inclinación del ojo (ver figura 14). Esta plantilla se ajusta a la imagen tratando de minimizar la energía, por lo tanto, se determinan las funciones de energía por potencial de valle, potencial de borde, potencial de imagen, potencial de pico y potencial interno. Los coeficientes se seleccionan para cada potencial y una regla de actualización se emplea para determinar el mejor conjunto de parámetros.

En los experimentos de los autores, se reporta que la localización inicial de la plantilla es crítica para determinar la localización exacta del ojo. Cuando la plantilla se inicio arriba de la ceja, el algoritmo falló en la distinción entre el ojo y la ceja. Por otra parte, la complejidad computacional de este algoritmo requiere de 5 a 10 minutos en un equipo SUN 4, una vez que se seleccionó un apropiado punto de inicio.

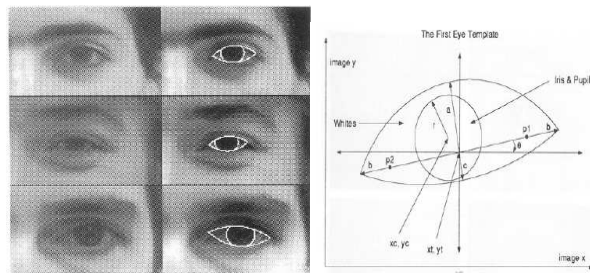


Figura 14: Sistema propuesto por Yuile, Cohen y Hallinan (1989), para el ojo.

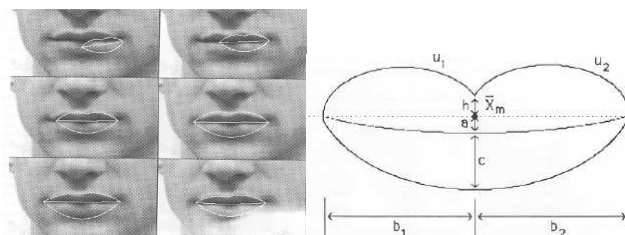


Figura 15: Sistema propuesto por Yuile, Cohen y Hallinan (1989), para la boca.

Nixon [25] utilizó la Transformada de Hough para lograr el reconocimiento facial. La transformada localiza formas descritas analíticamente al usar la magnitud del gradiente y la información direccional proporcionada por el operador de gradiente con el objeto de auxiliar en el proceso de reconocimiento.

Dos partes del ojo son atractivas para su reconocimiento, el perímetro redondeado del iris, porque la detección de formas circulares es relativamente sencilla. El perímetro de la esclera tiene la ventaja de que la forma es reflejada por la región que se encuentra debajo de las cejas.

La forma analítica que representa al iris es un círculo con dirección de gradiente esperadas en cada cuadrante, dada la luz de fondo de la esclera. Una elipse parece la forma más indicada para la representación aproximada de la forma del perímetro de la esclera, pero no es satisfactoria para aquellas partes alejadas del ojo a partir del centro de la cara (ver figura 16).

La elipse se forma para ojo de la cara utilizando un función exponencial. Los gradientes de magnitud que se obtuvieron mediante un operador de Sobel, se umbralizan utilizando cuatro niveles de brillos para representar la dirección del gradiente en ese punto. La información de la dirección se incorpora dentro de la técnica de la transformada de Hough.

El procedimiento para localizar cada ojo se restringe a la mitad de la imagen. La transformada de Hough se aplica para detectar la instancia de cada forma en un conjunto de datos de seis sujetos. La desviación de la posición del centro del iris a



partir del valor estimado tiene un valor medio de 0.33 píxeles. La aplicación de la transformada Hough para detectar el perímetro de la región debajo de las cejas parece en promedio, dar un espaciado de 20% mayor que el espaciado entre dos iris.

Utilizando la transformada Hough para encontrar la esclera, se muestra que el espaciado entre los ojos, difiere en promedio, por menos de 1.33 píxeles. Los resultados muestran que es posible derivar una medida del espaciado por la detección de la posición de ambos iris, la forma que describe el perímetro de la esclera y la cejas. La medida para detectar la posición del iris es más precisa. La detección del perímetro de la esclera es más sensitiva a los métodos.

La detección de la posición de las cejas proporciona una medida del espaciado del ojo, que es mayor que el proporcionado por las otras técnicas, pero que se puede utilizar cuando las otras no discriminan adecuadamente.

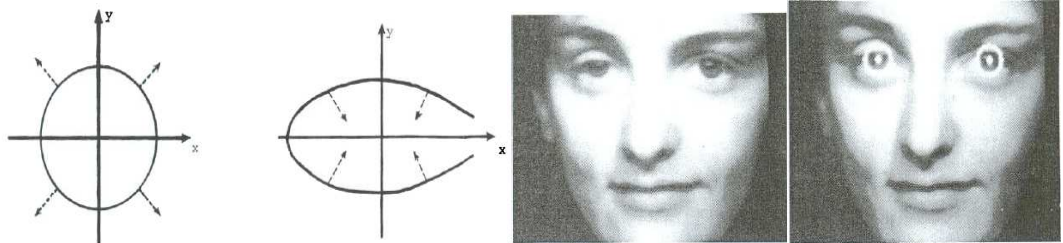


Figura 16: Sistema propuesto por Nixon (1985).

Manjunath y otros [25] presentan un método para la extracción de puntos característicos pertinentes a partir de la imagen de una cara (ver figura 17). Emplea la descomposición de Wavelet de Gabor y una interacción de escala local para extraer características en puntos de curvatura máxima en la imagen, correspondiente a la orientación y vecinos locales. Dichos puntos se almacenan en una base de datos y subsecuentes imágenes objetivo, se aparean utilizando una técnica de apareamiento de grafos.



Figura 17: Sistema propuesto por Manjunath y otros (1992).

**2.2.2.3. Reconocimiento de Rostros.** Kaya y Kobayashi [26] presentan un estudio básico en la clasificación de caras utilizando vistas frontales, con boca cerrada, sin barba y sin anteojos. Se utilizan distancias euclidianas entre puntos similares sobre la cara como parámetros, para la caracterización respectiva. Dichos parámetros son resistentes a cambio de luz, y pequeños cambios en la expresión facial. Estiman que el número de parámetros deben ser  $\log_2 N$ , en donde N es igual al número de caras a clasificar. Los parámetros se normalizan dividiéndose entre la longitud de la nariz, para contabilizar cualquier diferencia debida al tamaño de la fotografía y la distancia del sujeto a la cámara.

Se utilizaron fotos de 62 japoneses adultos entre los 20 y los 30 años de edad y bajo las mismas condiciones de luz. Los parámetros característicos fueron medidos a mano y se calculó la media y la desviación estándar. La correlación de los parámetros indica que la actual dimensión del vector de parámetros puede ser más pequeña que 9.

Unos de los métodos de caracterización de la cara es el uso de parámetros geométricos (distancias y ángulos entre los puntos, tales como las corneas del ojo, extremidades de la boca, mentón, etc.).

Los datos utilizados por Kanade [27], consisten en 17 hombres y 3 mujeres. Dos fotos de cada uno se tomaron, la segunda un mes después. Los puntos característicos se localizan en dos fases, un estado de “grueso a fino” simplifica la operación diferencial y los algoritmos para encontrar características. Una vez que los ojos, nariz y boca son aproximadamente localizados, una información más precisa se extrae al confinar el procesamiento a cuatro regiones pequeñas.

Estas cuatro regiones son los ojos izquierdo y derecho, nariz y boca (ver figura 18).

La intensidad se basa en el histograma de área local que se obtuvo en estado de “grueso a fino”.

Después de la adquisición fina de los datos de imagen, información más precisa se extrae de cada región utilizando un proceso de “thresholding” (umbral), diferenciación y proyección integral. Kanade utiliza un conjunto de 16 parámetros faciales que son proporciones de distancias, aéreas y ángulos, para compensar la variación de tamaño de las fotos. Para eliminar diferenciales de escala y dimensión de componentes del vector resultante se normalizan.

El conjunto entero de 40 imágenes se procesa y una foto de cada individuo se utiliza en el conjunto de referencia, las restantes 20 fotos se utilizan como conjunto de prueba. Se utilizó una simple medida de distancia para verificar la similitud entre una imagen del conjunto de prueba y una imagen en el conjunto de

referencia; se obtuvo una precisión entre el 45% y 75% dependiendo de los parámetros utilizados.

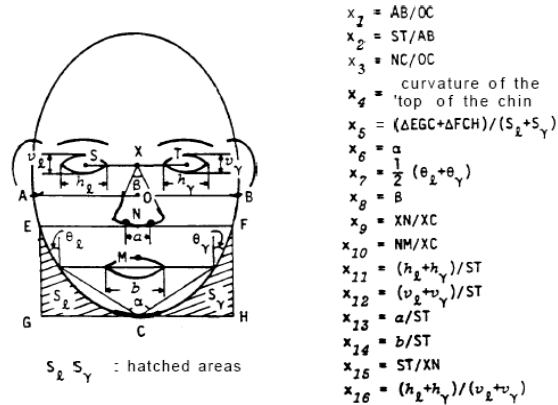


Figura 18: Sistema propuesto por Kanade (1977).

Recientemente, el uso de la expansión denominada “Karhunen-Loeve” KL (denominación que también se le ha dado a la técnica de análisis de componente principales), se ha utilizado para la representación de rostros por Kirby y Sirovich [28] y el reconocimiento de rostros por Turk y Pentland [2]. Dicha técnica ha generado nuevos intereses en investigación y en la actualidad, los dos principales sistemas de reconocimiento de rostros se basan en la técnica de componentes principales y sus variaciones. La expansión KL se ha utilizado para compresión de imágenes desde hace más de treinta años, su utilización en el reconocimiento de patrones se ha dado en pocas ocasiones (véase [29]).

Una de las razones por las cuales la expansión KL no se había utilizado para el reconocimiento de rostros, es su complejidad computacional.

Sirovich y Kirby [28] retoman el problema de representar imágenes mediante componentes principales, utilizando las caras sin orejas. Se puede notar el número de imágenes “M” para calcular la matriz de covarianza de los datos, es mucho menor que la dimensionalidad de renglones o columnas de dicha matriz, generando que la matriz sea singular. Los autores utilizan un método estándar del álgebra lineal que calcula únicamente M eigenvectores que no pertenecen al espacio nulo de la matriz degenerada. Una vez los eigenvectores (referidos como eigenfotos) se obtienen, cualquier imagen en el conjunto se puede reconstruir aproximadamente utilizando una combinación pesada de eigenfotos.

Turk y Pentland [2] utiliza la idea de los eigenfotos (conocidas como eigenfaces) para la detección de caras y su respectiva identificación. Dadas las eigenfaces (ver figura 19), cualquier cara en la base de datos se puede representar como un vector de pesos, los pesos se obtienen por la proyección de la imagen en los componentes de la eigenface por medio de un simple producto punto.

Cuando una imagen va a ser reconocida, también se representa por su vector de pesos. La identificación de la imagen de prueba se realiza al localizar la imagen en la base de datos cuyos pesos son los más cercanos (en distancia Euclídea), a los pesos de la imagen de prueba. Se reporta que el sistema es robusto para cambios en la condición de luz, pero se degrada rápidamente ante cambios de escala.

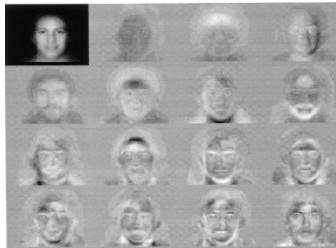


Figura 19: Sistema propuesto por Turk y Pentland (1991).

Brunelli y Poggio [30], desarrollaron un método para el reconocimiento de rostro a partir de características geométricas del rostro (ver figura 20). A partir de 35 características se realiza el reconocimiento mediante un clasificador bayesiano.

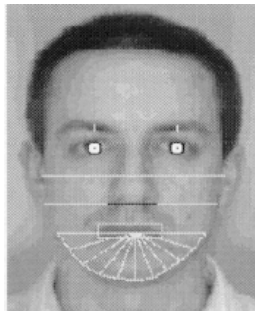


Figura 20: Sistema propuesto por Brunelli y Poggio (1995).

Otro método, muy utilizado hasta la fecha, consiste en la utilización de “plantillas” (mascaras), en el cual se configuran diversas plantillas que abarcan ciertas regiones del rostro: ojos, nariz, boca, etc. (ver figura 21). La imagen a reconocer se compara a través de las plantillas con las imágenes de la base de datos, y se obtiene un vector con la puntuación de apareamiento mediante procesos de correlación [31].



Figura 21: Sistema propuesto por Brunelli y Poggio (1994).

Lucas [32] propone el método denominado Clasificador de n-tuplas (ver figura 22), el cual consiste en la selección de píxeles por medio de un muestreo de “n-tuplas” de estos (obteniendo únicamente los valores de nivel de gris). Cada tupla presenta una longitud de “m”. El reconocimiento se efectúa mediante la utilización de una métrica (distancia Manhattan), para comparar el conjunto de entrenamiento con la imagen a clasificar.

La utilización de redes neuronales en el reconocimiento de caras se ha dirigido a diversos problemas: clasificación de géneros, reconocimiento de la cara, clasificación de la expresión facial.

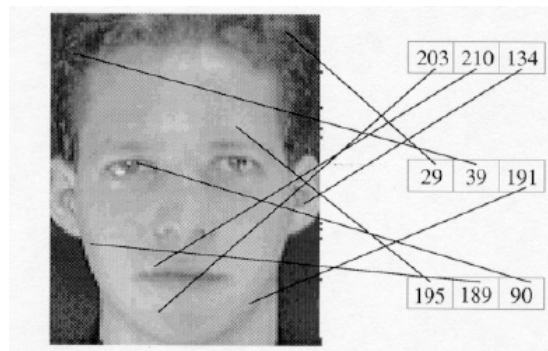


Figura 22: Sistema propuesto por Lucas (1995).

Una de las más antiguas aplicaciones se reporta en el *mapa asociativo* de Kohonen [33], utilizando un pequeño conjunto de caras, se reporta una precisión cuando la imagen de entrada presenta bastante “ruido” o cuando porciones de la imagen se omiten.

Una simple capa adaptativa (una por cada persona en la base de datos) para reconocimiento de caras, análisis de expresión facial y verificación de cara, se reporta en un estudio de Stonham [34]. Este sistema se denominó WISARD. El sistema necesita de 200 a 400 presentaciones para entrenamiento de cada

clasificador, los patrones de entrenamiento incluyen translación y variación en expresión facial. 16 clasificadores se utilizaron para construir el conjunto de datos utilizando 16 personas. La clasificación se logró, determinando el clasificador que proporciona la más alta respuesta para la imagen de entrada. Sin embargo, el reducido tamaño de la muestra no permite realizar conclusiones adecuadas sobre la viabilidad de este enfoque aplicado a grandes conjuntos de personas en la base de datos.

Utilizando un vector de 16 atributos numéricos, tales como ancho de la ceja, ancho de la nariz y boca, etc., Brunelli y Poggio [35], desarrollaron una red neuronal para la clasificación de género. Los autores entrenaron dos redes, una para cada tipo de género. Las imágenes de entrada se normalizaron respecto a la escala y rotación, utilizando la posición de los ojos, los cuales son detectados de manera automática. El vector de características de 16 dimensiones, se extrae también de manera automática. Las salidas de las dos redes se comparan, la etiqueta de género para la imagen de prueba se decide por la red con mayor salida. En experimentos de clasificación, únicamente un subconjunto de vector de 16 características, se utilizó. La base de datos comprendía una imagen del rostro de 21 hombres y 21 mujeres.

La estrategia "leave-one-out" [36], se utilizó para la clasificación.

Cuando el vector de características se utilizó como vector de prueba se logró una precisión del 92.5%. Para caras que no se encontraban en el conjunto de entrenamiento, la precisión fue del 87.5%.

Seibert y Waxman [37], han propuesto un sistema para reconocimiento de caras a partir de sus partes utilizando una red neuronal. El sistema es similar a un sistema modular desarrollado para reconocer objetos en 3-D, a partir de combinación de las vistas en 2-D desde diversos puntos, en el caso de las caras, los arreglos de características tales como ojos y nariz tienen un papel importante en las vistas en 2-D. los pasos del procesamiento son: segmentación de la región de la cara utilizando técnicas de detección de cambio entre frames, extracción de características como los ojos, boca, etc., utilizando la detección de simetría, agrupamiento y mapeo log-polar de las características y sus atributos tales como centroides, codificación de arreglos de características, *clustering* de vectores de características en categorías de visión utilizando ART 2, e integración de evidencia acumulada utilizando una red de aspecto.

**2.2.2.4. Métodos evolutivos.** Liu y Wechsler [38], proponen una combinación de Análisis de componentes principales con un enfoque evolutivo. Se utiliza un Algoritmo Genético con el propósito de buscar los mejores ejes de rotación definidos por el Análisis de Componentes Principales, con el propósito de encontrar los ejes que mejor realicen la separación de clases (ver figura 23).

Reportan experimentos que se realizaron con 1107 imágenes de rostros de la base de datos FERET, correspondientes a 369 sujetos, utilizando en promedio, tres imágenes por sujeto.

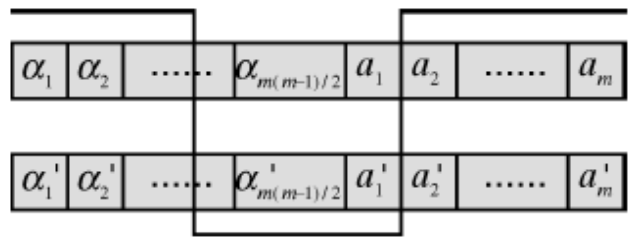


Figura 23: Sistema propuesto por Liu y Wechsler (1998).

Otras aplicaciones utilizando algoritmos genéticos, se han realizado combinándolos con algunas de las técnicas mencionadas en las secciones anteriores.

Entre las investigaciones reportadas se encuentran las de Bhanu [39] para segmentación de imágenes. Huang [40], realiza reconocimiento utilizando componentes principales y algoritmos genéticos. Pinto y Sossa [41], proponen un método que utiliza las primeras cuatro invariantes de Hu y a partir de estas, realizar la identificación la identificación con ayuda de un algoritmo genético.

## 2.3. CONCEPTOS DE ESTADISTICA

**2.3.1. Media.** La media es una medida de tendencia central.

La media de las observaciones  $x_1, x_2, x_3, \dots, x_n$  es el promedio de las observaciones. Y se denota:

$$\bar{x} = \frac{x_1 + x_2 + \dots + x_n}{n} = \frac{\sum_{i=1}^n x_i}{n} \quad (2.3.1)$$

Por ejemplo, para los valores de la variable  $X = \{1, 2, 7\}$  la media  $X$  seria = 3.3. (Ver figura 24).



Figura 24: Interpretación geométrica de la media en una variable.

Si fueran dos variables  $\langle X, Y \rangle$ , la media se calcula individualmente por cada variable.

X	Y
1	2
2	5
7	3

La media de  $X = 3.3$ , la media de  $Y = 3.3$ . (ver figura 25).

Tabla 1: Ejemplo de datos de dos variables.

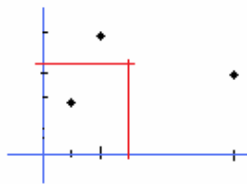


Figura 25: Interpretación geométrica de la media en dos variables.

Se puede interpretar geoméricamente la media como el “centro de gravedad de los datos”.

Se dice que los datos están centrados cuando su media es cero. Esto se puede lograr restando a cada valor de la variable, la media de estos.



Geoméricamente esto es equivalente a trasladar el “centro de gravedad” al origen y trasladar cada dato en el proceso. (Ver figura 26).

$$x_i' = x_i - \bar{x} \quad (2.3.2)$$

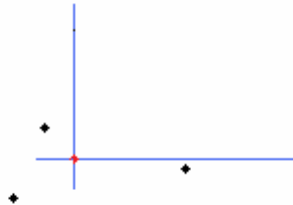


Figura 26: interpretación geométrica de datos centrados.

**2.3.2. Varianza.** Cuando se tiene una variable, la varianza mide la variabilidad de los datos.

Geoméricamente es el promedio de las distancias desde la media hasta los datos.

La formula es:

$$S_x^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x}) \cdot (x_i - \bar{x}) \quad (2.3.3)$$

Cuando se tienes dos variables como en el ejemplo anterior, se puede calcular a la varianza a cada variable, así:

$$S_x^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x}) \cdot (x_i - \bar{x}) \quad (2.3.4)$$

$$S_y^2 = \frac{1}{n} \sum_{i=1}^n (y_i - \bar{y}) \cdot (y_i - \bar{y}) \quad (2.3.5)$$

En el ejemplo

$$S_x^2 = 6.89 \quad S_y^2 = 1.55$$

Observar la siguiente gráfica (figura 27):

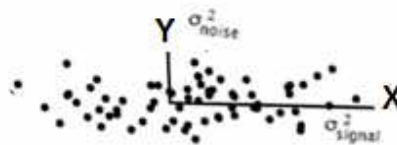


Figura 27: Interpretación de la varianza como el nivel de señal.

En este caso la varianza en la dirección de X se puede interpretar como el nivel de señal de la variable, y la varianza en la dirección de Y como el nivel del ruido.

A la relación:

$$SNR = \frac{\sigma_{signal}^2}{\sigma_{noise}^2} \quad (2.3.6)$$

Se la conoce como tasa de señal-ruido. Por lo que una alta varianza en X, hace un nivel señal-ruido mayor, esto significa que los datos tienen menos introducción de errores (señales ruidosas).

**2.3.3. Covarianza.** La covarianza es una generalización de la varianza y mide el grado de correlación de dos variables.

La covarianza de X y Y se define como:

$$\text{Cov}(X,Y) = S_{xy}^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x}) \cdot (y_i - \bar{y}) \quad (2.3.7)$$

Se puede ver que la varianza es un caso particular de covarianza donde  $\text{var}(x) = \text{Cov}(x,x)$ .

### 2.3.4. Interpretación geométrica de varianza y covarianza.

Si se tienen las observaciones de las variables

$$(x_i, y_i), \quad i=1,2,\dots,n$$

Se puede escribir las observaciones en forma de vectores

$$\vec{X} = (x_1, x_2, \dots, x_n) \in R^n \quad (2.3.8)$$

$$\vec{Y} = (y_1, y_2, \dots, y_n) \in R^n \quad (2.3.9)$$

$$\vec{\bar{x}} = (\bar{x}, \bar{x}, \dots, \bar{x}) \in R^n \quad (2.3.10)$$

$$\vec{\bar{y}} = (\bar{y}, \bar{y}, \dots, \bar{y}) \in R^n \quad (2.3.11)$$

Si se denota  $\vec{v} \cdot \vec{w}$  al producto escalar de los vectores  $\vec{v}$  y  $\vec{w}$ , es inmediato

comprobar que en realidad las definiciones de varianza y covarianza tienen una idea geométrica muy simple: son productos escalares en los que intervienen los vectores  $\vec{X} - \vec{\bar{x}}$  e  $\vec{Y} - \vec{\bar{y}}$

$$s_{xy} = \frac{1}{n}(\vec{X} - \bar{x}) \cdot (\vec{Y} - \bar{y}) \quad (2.3.12)$$

$$s_x^2 = \frac{1}{n}(\vec{X} - \bar{x}) \cdot (\vec{X} - \bar{x}) = \frac{1}{n}|\vec{X} - \bar{x}|^2 = s_{xx} \quad (2.3.13)$$

$$s_y^2 = \frac{1}{n}(\vec{Y} - \bar{y}) \cdot (\vec{Y} - \bar{y}) = \frac{1}{n}|\vec{Y} - \bar{y}|^2 = s_{yy} \quad (2.3.14)$$

Con esta descripción geométrica de las varianzas y covarianzas, se pone en manifiesto la existencia de paralelismo entre las desviaciones de las variables X e Y, con respecto a sus centros de gravedad ya que

$$(\vec{X} - \bar{x}) \cdot (\vec{Y} - \bar{y}) = |\vec{X} - \bar{x}| \cdot |\vec{Y} - \bar{y}| \cdot \cos \theta \quad (2.3.15)$$

donde  $\theta$  es el ángulo entre los vectores  $\vec{X} - \bar{x}$  e  $\vec{Y} - \bar{y}$  (véase la figura 28).

Despejando:

$$\cos \theta = \frac{(\vec{X} - \bar{x}) \cdot (\vec{Y} - \bar{y})}{|\vec{X} - \bar{x}| \cdot |\vec{Y} - \bar{y}|} = \frac{\frac{1}{n}(\vec{X} - \bar{x}) \cdot (\vec{Y} - \bar{y})}{\sqrt{\frac{1}{n}}|\vec{X} - \bar{x}| \cdot \sqrt{\frac{1}{n}}|\vec{Y} - \bar{y}|} = \frac{s_{xy}}{s_x \cdot s_y} \quad (2.3.16)$$

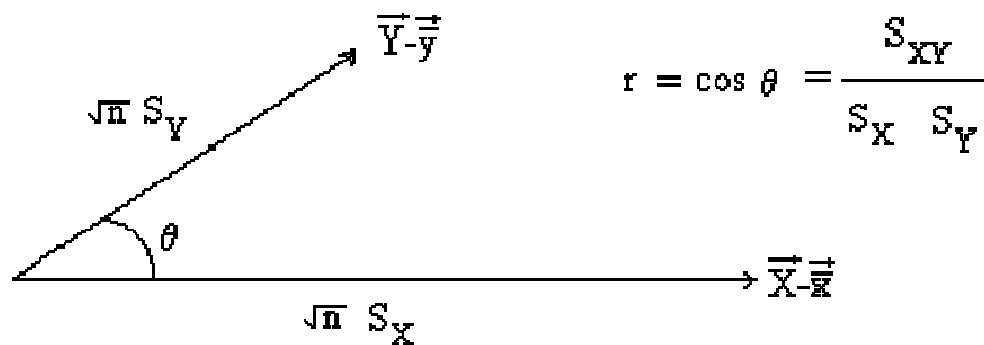


Figura 28: Interpretación geométrica de  $r$  como el coseno del ángulo que forman los vectores de las desviaciones con respecto a sus respectivas medias de X y de Y.

Si los vectores  $\vec{X} - \bar{x}$  e  $\vec{Y} - \bar{y}$  son totalmente paralelos entonces  $\cos \theta = \pm 1$  (ver figura 29). En este caso existirá una constante de proporcionalidad  $m$  tal que:

$$(\vec{Y} - \bar{y}) = m \cdot (\vec{X} - \bar{x})$$

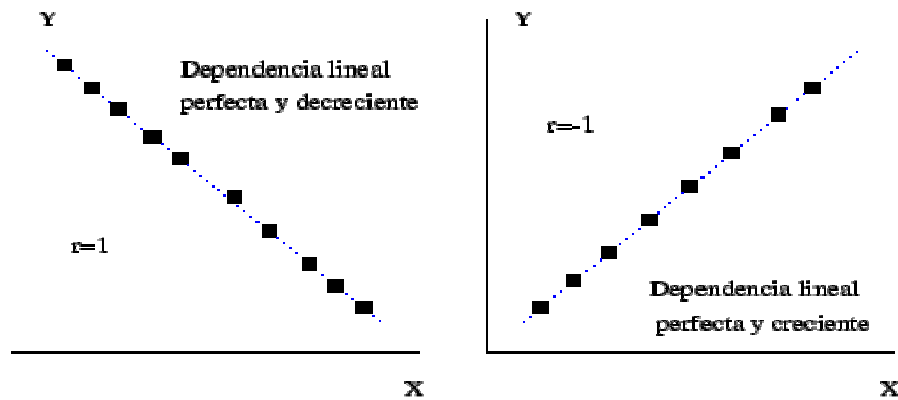


Figura 29: Dependencia lineal de dos variables.

En el ejemplo anterior, el que se pueda determinar  $Y = mX + y$ , es un ejemplo de redundancia de las variables, ya que no es necesario dos variables  $(x,y)$  para representar los datos, sino, solamente una  $(x,mx)$ .

Por lo tanto, la covarianza es un buen indicador de la redundancia de la información, una covarianza de cero indica que no hay redundancia o que los datos están des correlacionados.

Se pueden formar todos los caso de covarianza:  $Cov(x,x)$ ,  $Cov(x,y)$ ,  $Cov(y,x)$ ,  $Cov(y,y)$  y agruparlos en una matriz; la Matriz de Covarianza

$$C = \begin{bmatrix} Cov(x,x) & Cov(x,y) \\ Cov(y,x) & Cov(y,y) \end{bmatrix} = \begin{bmatrix} S_X^2 & S_{XY} \\ S_{YX} & S_Y^2 \end{bmatrix} \quad (2.3.17)$$

Si se tiene  $n$  variables, la matriz de covarianza será una matriz de  $n \times n$ .

### 2.3.5. Propiedades de la matriz de covarianza.

- Es una matriz real simétrica (cuadrada).

- Los elementos de la diagonal son las varianzas de las variables. Nivel señal de la variable.
- Los elementos fuera de la diagonal son las covarianzas: grados de interrelación de una variable  $i$  con respecto a una variable  $j$ .

## 2.4. CONCEPTOS DE ALGEBRA LINEAL

**2.4.1. Matriz diagonal.** Es una matriz cuyos elementos que no pertenecen a la diagonal son ceros.

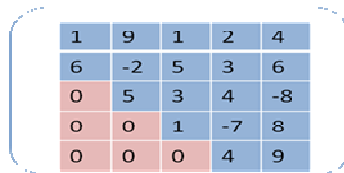
Un caso particular de matriz diagonal es la matriz unitaria, en el cual los elementos que no pertenecen a la diagonal son cero y los de la diagonal son unos.

Ej:

$$A = \begin{bmatrix} 4 & 0 & 0 \\ 0 & -3.2 & 0 \\ 0 & 0 & 19.50 \end{bmatrix} \text{ (matriz diagonal)}$$

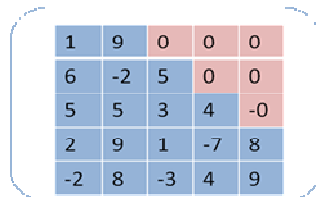
$$B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ (matriz unitaria)}$$

**2.4.2. Matriz de Hessenberg.** Es una matriz en la cual los elementos debajo de la subdiagonal son ceros. (Ver figura 30 y 31).



1	9	1	2	4
6	-2	5	3	6
0	5	3	4	-8
0	0	1	-7	8
0	0	0	4	9

Figura 30: Matriz de Hessenber inferior.



1	9	0	0	0
6	-2	5	0	0
5	5	3	4	-0
2	9	1	-7	8
-2	8	-3	4	9

Figura 31: Matriz de Hessenber superior.

**2.4.3. Matriz tridiagonal.** Una matriz tridiagonal es aquella cuyos elementos que no están ni en la diagonal principal ni en las subdiagonales superiores e inferiores son cero. (Ver figura 32).

1	9	0	0	0
6	-2	5	0	0
0	5	3	4	-0
0	0	1	-7	8
0	0	0	4	9

Figura 32: Matriz tridiagonal.

**2.4.4. Base de un espacio.** Sea  $E$  un espacio, una base  $B$  es un subespacio de  $E$ , que permite construir objetos de dicho espacio.

Por ejemplo la base  $B$  del espacio  $E=R^2$ ,

$B = \{p_1, p_2\} = \{ (1,0), (0,1) \}$  permite construir cualquier objeto en  $R^2$ .

Para construir  $(3,2) = 3p_1 + 2p_2 = 3(1,0) + 2(0,1) = (3,0) + (0,2) = (3,2)$

**2.4.5. Cambio de base.** Así como  $B$  es base para construir  $R^2$ , también lo puede ser  $B' = \{p'_1, p'_2\} = \{ (0.866, 0.5), (-0.5, 0.866) \}$

$$p'_1 = 0.866p_1 + 0.5p_2$$

$$p'_2 = -0.5p_1 + 0.866p_2$$

Se puede observar que la nueva base es una combinación lineal de la base original.

Se puede construir una matriz de cambio de base  $U$  con los coeficientes anteriores, que permite pasar de la base  $B$  a  $B'$

$$U = \begin{bmatrix} 0.866 & 0.5 \\ -0.5 & 0.866 \end{bmatrix}$$

La relación entre la base  $B$  y  $B'$  esta dada por la siguiente relación

$$B' = U^T B \tag{2.4.1}$$

De la misma forma el objeto  $(3,2)$  en la base  $B$ , es equivalente a  $(3.232, 1.598)$  en la base  $B'$ :

$$(x,y) = \begin{bmatrix} 0.866 & 0.5 \\ -0.5 & 0.866 \end{bmatrix} \cdot \begin{bmatrix} 3 \\ 2 \end{bmatrix} = \begin{bmatrix} 3.232 \\ 1.598 \end{bmatrix} = (3.232, 1.598)$$

En el ejemplo la rotación es de 30 grados en dirección contraria las manecillas del reloj.

La siguiente figura muestra lo anterior, geoméricamente. (Figura 33).

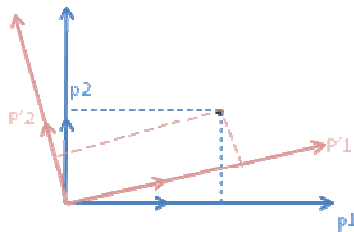


Figura 33: interpretación geométrica de U, como una rotación de los ejes p1,p2 en p'1,p'2.

**2.4.6. Vectores y valores propios.** Sea  $\mathbf{v}$  un vector, y  $\mathbf{A}$  una matriz, se dice que  $\mathbf{v}$  es un “vector propio” de  $\mathbf{A}$  si:

$$\mathbf{A} \cdot \mathbf{v} = \alpha \mathbf{v} \tag{2.4.2}$$

Es decir que si al efectuar la transformación  $\mathbf{A}$  sobre el vector  $\mathbf{v}$ , este es el mismo vector multiplicado por un múltiplo  $\alpha$ , al cual se llama “valor propio”.

Ejemplo:

$$\mathbf{A} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

$$\mathbf{v} = [1, 1]$$

$$\mathbf{A} \cdot \mathbf{v} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \end{bmatrix} = 3 \times \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$\alpha = 3$ , sería el valor propio

El siguiente ejemplo muestra como calcular los vectores y valores propios “a mano” para una matriz de 2x2

Sea  $\mathbf{A} = \begin{bmatrix} -5 & 2 \\ 2 & -2 \end{bmatrix}$ , se desea encontrar el vector propio  $\mathbf{x}$  (no nulo) y su valor propio  $\alpha$ .



**Determinar el valor propio  $\alpha$ .**

$$\mathbf{A}\mathbf{x} = \alpha\mathbf{x} \quad (2.4.3)$$

$$\mathbf{A}\mathbf{x} - \alpha\mathbf{x} = 0$$

Aunque parece tentativo factorizar  $\mathbf{x}$ , se nota que  $\mathbf{A}$  es una matriz y  $\alpha$  es un escalar; ambos tienen que ser matrices. La expresión  $\alpha\mathbf{x}$  (que es un vector) es equivalente a multiplicar por la matriz unitaria o identidad  $\mathbf{I}$  por  $\alpha\mathbf{x}$ , obteniéndose  $\alpha\mathbf{I}\mathbf{x}$ . Ahora ya se tiene una matriz al lado derecho y se puede factorizar.

$$\mathbf{A}\mathbf{x} - \alpha\mathbf{I}\mathbf{x} = 0$$

$$(\mathbf{A} - \alpha\mathbf{I})\mathbf{x} = 0 \quad (2.4.4)$$

Como  $\mathbf{x}$ , es por definición un vector no nulo, se puede eliminar

$$(\mathbf{A} - \alpha\mathbf{I}) = 0 \quad (2.4.5)$$

$$\begin{bmatrix} -5 & 2 \\ 2 & -2 \end{bmatrix} - \alpha \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = 0$$

$$\begin{bmatrix} -5 - \alpha & 2 \\ 2 & -2 - \alpha \end{bmatrix} = 0$$

Este sistema tiene solución si el determinante es cero

$$D(\alpha) = \det(\mathbf{A} - \alpha\mathbf{I}) = 0 \quad (2.4.6)$$

$$\begin{vmatrix} -5 - \alpha & 2 \\ 2 & -2 - \alpha \end{vmatrix} = (-5 - \alpha)(-2 - \alpha) - 4 = 0$$

$$\alpha^2 + 7\alpha + 6 = 0, \quad (\alpha+6)(\alpha+1)=0, \quad \alpha = -1, -6$$

**Determinar el vector propio  $\mathbf{x}$**

Si se toma el primer valor propio  $\alpha=-1$

de la ecuación (2.4.4)

$$\begin{bmatrix} -4 & 2 \\ 2 & -1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$$

$$-4x_1 + 2x_2 = 0$$

$$2x_1 - 1x_2 = 0$$

$$x_2 = 2 x_1 \quad \mathbf{x} = (1,2)$$

Comprobando

$$\mathbf{A}\mathbf{x} = \begin{bmatrix} -5 & 2 \\ 2 & -2 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} -1 \\ -2 \end{bmatrix} = -1 \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

Se observa que calcular los vectores y valores propios por medio de la anterior técnica es muy compleja para implementarla en un algoritmo de computadora, y más aún cuando las matrices son más grandes, es por ello, que en la siguiente sección se estudiará técnicas mucho más efectivas para el cálculo de vectores propios.

**2.4.7. Diagonalización de matrices reales simétricas.** El proceso de diagonalización de una matriz consiste (en este caso, una matriz real simétrica), en encontrar una transformación T que al aplicarla a una matriz A, la convierta en diagonal.

La matriz diagonal resultante son los Eigenvalores de A y la matriz T son los Eigenvectores de A.

Aplicar sucesivas rotaciones de Jacobi (que se verá a continuación) garantiza que se encontrarán todos los eigenvectores de A; estos aparecen en las columnas de T, pero el método es computacionalmente intenso.

La gran estrategia consiste en convertir previamente la matriz A en forma tridiagonal a través de reducción de Householder, y luego aplicar rotaciones de Jacobi a la subdiagonal de la matriz resultante, de esta forma, después de iteraciones sucesivas los eigenvectores aparecen en las columnas de T [42].

Primero se verá la rotación de Jacobi, que es más sencilla e intuitiva, y después la rotación y reducción de Householder.

**2.4.7.1. Rotación de Jacobi.** La rotación de Jacobi, es utilizada para transformar un elemento i,j de una matriz a cero [42],

Por ejemplo si tiene la matriz A:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \tag{2.4.7}$$

Si se desea transformar el elemento  $a_{21}$  a cero, la matriz de rotación de Jacobi será:

$$R = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.4.8)$$

El efecto de aplicar las siguientes transformaciones a la matriz A se muestra a continuación:

$$R^T A R = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.4.9)$$

Primero se multiplica AR

$$AR = \begin{bmatrix} a_{11} \cos \theta - a_{12} \sin \theta & a_{11} \sin \theta + a_{12} \cos \theta & a_{13} \\ a_{21} \cos \theta - a_{22} \sin \theta & a_{21} \sin \theta + a_{22} \cos \theta & a_{23} \\ a_{31} \cos \theta - a_{32} \sin \theta & a_{31} \sin \theta + a_{32} \cos \theta & a_{33} \end{bmatrix} \quad (2.4.10)$$

A este resultado se pre multiplica por  $R^T$

$$R^T A R = \quad (2.4.11)$$

$$\begin{bmatrix} a_{11} \cos^2 \theta + a_{22} \sin^2 \theta - 2a_{12} \cos \theta \sin \theta & a_{12}(\cos^2 \theta - \sin^2 \theta) + (a_{11} - a_{22}) \cos \theta \sin \theta & a_{13} \cos \theta - a_{23} \sin \theta \\ a_{12}(\cos^2 \theta - \sin^2 \theta) + (a_{11} - a_{22}) \cos \theta \sin \theta & a_{22} \cos^2 \theta + a_{11} \sin^2 \theta + (a_{12} + a_{21}) \cos \theta \sin \theta & a_{13} \sin \theta + a_{23} \cos \theta \\ a_{31} \cos \theta - a_{32} \sin \theta & a_{32} \cos \theta + a_{13} \sin \theta & a_{33} \end{bmatrix}$$

Si se quiere anular el elemento  $a'_{12}$  se debe elegir el ángulo de rotación  $\theta$ , que cumple

$$a_{12}(\cos^2 \theta - \sin^2 \theta) + (a_{11} - a_{22}) \cos \theta \sin \theta = 0 \quad (2.4.12)$$

$$\frac{a_{12}}{a_{22} - a_{11}} = \frac{\cos \theta \sin \theta}{\cos^2 \theta - \sin^2 \theta} \quad (2.4.13)$$

Se puede expresar la igualdad en función del ángulo doble:

$$\alpha = \cot 2\theta = \frac{a_{22} - a_{11}}{2a_{12}} \quad (2.4.14)$$

Se ve que no es necesario invocar funciones trigonométricas para calcular  $\cot 2\theta$ .

A fin de economizar tiempo de cálculo es también conveniente calcular  $\sin \theta$  y  $\cos \theta$

$$\alpha = \cot 2\theta = \frac{\cos^2\theta - \sin^2\theta}{2\cos\theta\sin\theta} = \frac{a_{22} - a_{11}}{2a_{12}} \quad (2.4.15)$$

$$\cot 2\theta = \frac{\cos^2\theta - \sin^2\theta}{2\cos\theta\sin\theta} = \frac{1 - \tan^2\theta}{2\tan\theta} \quad (2.4.16)$$

Como ya se ha expresado, si hace  $\cot 2\theta = \alpha$

$$\tan^2\theta + 2\alpha\tan\theta - 1 = 0 \quad (2.4.17)$$

$$\tan\theta = -\alpha \pm \sqrt{\alpha^2 + 1} \quad (2.4.18)$$

En este caso, es conveniente la rotación más pequeña

$$\tan\theta = -\alpha + \text{sgn}(\alpha)\sqrt{\alpha^2 + 1}$$

De esta forma, se puede pasar a una rotación de Jacobi  $\mathbf{P}$  en su forma más general, dicha rotación eliminará el elemento  $a_{pq}$ .

Una matriz de rotación  $\mathbf{P}$  tendrá 1 en la diagonal y 0 fuera de la diagonal, excepto en los elementos  $P_{pp}$ ,  $P_{qq}$ ,  $P_{pq}$ ,  $P_{qp}$

$$\mathbf{P} = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & P_{pp} & \cdots & P_{pq} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & P_{qp} & \cdots & P_{qq} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix} = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cos\theta & \cdots & \sin\theta & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & -\sin\theta & \cdots & \cos\theta & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix}$$

donde definiendo

$$\alpha = \frac{a_{qq} - a_{pp}}{2a_{pq}}$$

La anulación de  $a'_{pq}$  proporciona las siguientes relaciones para  $\sin\theta$ ,  $\cos\theta$ ,  $\tan\theta$

$$\tan\theta = -\alpha + \text{sgn}(\alpha)\sqrt{\alpha^2 + 1}$$

$$\cos\theta = \frac{1}{\sqrt{\tan^2\theta+1}} \quad (2.4.19)$$

$$\sin\theta = \cos\theta \tan\theta \quad (2.4.20)$$

Mediante esta rotación, denominando  $s = \sin \theta$ ,  $c = \cos \theta$  y  $t = \tan \theta$ , y teniendo en cuenta la ecuación (2.4.11), los elementos de la matriz  $A$  se transforman como

$$a_{pp} = c^2 a_{pp} + s^2 a_{qq} - 2sc a_{pq} \quad (2.4.21)$$

$$a_{qq} = c a_{qq} + s^2 a_{pp} + 2sc a_{pq} \quad (2.4.22)$$

$$a_{pq} = (c^2 - s^2) a_{pq} + sc (a_{pp} - a_{qq}) \quad (2.4.23)$$

$$a_{rp} = c a_{rp} - s a_{rq} \quad (2.4.24)$$

$$a_{rq} = c a_{rq} + s a_{rp} \quad (2.4.25)$$

Donde  $r$  indica todos los índices distintos de  $p$  y  $q$ . Todos los demás elementos quedan inalterados.

Estas relaciones se pueden escribir de forma aún más compacta, teniendo en cuenta la anulación de  $a_{pq}$ , como:

$$a_{pp} = a_{pp} - t a_{pq} \quad (2.4.26)$$

$$a_{qq} = a_{qq} + t a_{pq} \quad (2.4.27)$$

$$a_{rp} = a_{rp} - s(a_{rq} + a_{rp}) \quad (2.4.28)$$

$$a_{rq} = a_{rq} + s(a_{rp} - a_{rq}) \quad (2.4.29)$$

**2.4.7.2. Rotación de Householder.** La rotación de Householder será utilizada próximamente para reducir una matriz cuadrada simétrica a una matriz tridiagonal [43].

La rotación de Householder  $\mathbf{P}$  se define de la siguiente forma:

$$\mathbf{P} = \mathbf{1} - 2 \mathbf{w} \otimes \mathbf{w} \quad (2.4.30)$$

Donde  $\mathbf{1}$  es una matriz unitaria y  $\mathbf{w} \otimes \mathbf{w}$  es el producto tensorial de los vectores  $\mathbf{w}$  y  $\mathbf{w}$ .

El producto tensorial  $\mathbf{w} \otimes \mathbf{w}$  se define como  $\mathbf{w}_i \mathbf{w}_j$ .

A diferencia del producto escalar de dos vectores que es un número o el producto cruz de dos vectores, que es un vector, el producto tensorial de dos vectores es una matriz y se logra multiplicando cada elemento del primer vector por todos los elementos del segundo.

Ej: si  $\mathbf{x} = (2, 1, 3)$

$$x \otimes x = \begin{bmatrix} 2x_2 & 2x_1 & 2x_3 \\ 1x_2 & 1x_1 & 1x_3 \\ 3x_2 & 3x_1 & 3x_3 \end{bmatrix} = \begin{bmatrix} 4 & 2 & 6 \\ 2 & 1 & 3 \\ 6 & 3 & 9 \end{bmatrix}$$

Puede verse que el producto tensorial de un vector por si mismo da como resultado una matriz simétrica. Por lo que  $\mathbf{P}$  también es simétrico:

$$\mathbf{P}^T = \mathbf{P} \quad (2.4.31)$$

En la ecuación (2.4.30) la única restricción que se pide es que  $\mathbf{w}$  sea unitario, esto es:

$$\mathbf{w} \cdot \mathbf{w} = 1 \quad (2.4.32)$$

Otra propiedad de  $\mathbf{P}$  es que su inversa es igual a  $\mathbf{P}$  :

$$\begin{aligned} \mathbf{P} \cdot \mathbf{P} &= (1 - 2 \mathbf{w} \otimes \mathbf{w}) \cdot (1 - 2 \mathbf{w} \otimes \mathbf{w}) \\ &= 1 \cdot 1 - 1 \cdot 2 \mathbf{w} \otimes \mathbf{w} - 2 \mathbf{w} \otimes \mathbf{w} \cdot 1 + 4(\mathbf{w} \otimes \mathbf{w}) \cdot (\mathbf{w} \otimes \mathbf{w}) \\ &= \dots \end{aligned}$$

donde

$$(\mathbf{w} \otimes \mathbf{w}) \cdot (\mathbf{w} \otimes \mathbf{w}) = \sum w_i w_j w_j w_k = w_i w_k = \mathbf{w} \otimes \mathbf{w}$$

$$\mathbf{P} \cdot \mathbf{P} = 1 - 4 \mathbf{w} \otimes \mathbf{w} + 4 \mathbf{w} \otimes \mathbf{w} = 1$$

en resumen  $\mathbf{P} = \mathbf{P}^T = \mathbf{P}^{-1}$ , así que  $\mathbf{P}$  es ortogonal

Se puede usar cualquier vector  $\mathbf{u}$  en lugar de  $\mathbf{w}$  si es normalizado al mismo tiempo

$$\mathbf{P} = 1 - \frac{2\mathbf{u} \otimes \mathbf{u}}{\mathbf{u} \cdot \mathbf{u}} = 1 - \frac{\mathbf{u} \otimes \mathbf{u}}{H} \quad (2.4.33)$$

Donde

$$H = \frac{\mathbf{u} \cdot \mathbf{u}}{2} \quad (2.4.34)$$

Si se escoge  $\mathbf{u}$  de la siguiente forma:

$$\mathbf{u} = x \bar{\mathbf{r}} + |x| \mathbf{e}_1 \quad (2.4.35)$$

Donde  $\mathbf{e}_1$  es el primer vector base

Con la definición del vector  $\mathbf{u}$  de esta forma, el operador de Householder tiene el efecto de rotar el vector  $\mathbf{x}$  en la dirección de  $\mathbf{e}_1$ , esto se lo puede ver de la siguiente forma (ver figura 34):

$$\mathbf{P} \cdot \mathbf{x} = \left( \mathbf{1} - \frac{\mathbf{u} \otimes \mathbf{u}}{H} \right) \cdot \mathbf{x} \quad (2.4.36)$$

$$= \left( \mathbf{1} - \frac{2\mathbf{u} \otimes (\mathbf{x} \mp |\mathbf{x}| \mathbf{e}_1)}{\mathbf{u} \cdot \mathbf{u}} \right) \cdot \mathbf{x} \quad (2.4.37)$$

Donde  $\mathbf{u} \cdot \mathbf{u}$  es igual a:

$$\begin{aligned} \mathbf{u} \cdot \mathbf{u} &= (\mathbf{x} \mp |\mathbf{x}| \mathbf{e}_1) \cdot (\mathbf{x} \mp |\mathbf{x}| \mathbf{e}_1) \\ &= \mathbf{x} \cdot \mathbf{x} \mp 2|\mathbf{x}| \mathbf{e}_1 \cdot \mathbf{x} + |\mathbf{x}|^2 \mathbf{e}_1 \cdot \mathbf{e}_1 \\ &= |\mathbf{x}|^2 \mp 2|\mathbf{x}|x_1 + |\mathbf{x}|^2 \\ &= 2|\mathbf{x}|^2 \mp 2|\mathbf{x}|x_1 \end{aligned} \quad (2.4.38)$$

Sustituyendo este último en la ecuación (2.4.37)

$$\begin{aligned} \mathbf{P} \cdot \mathbf{x} &= \left( \mathbf{1} - \frac{2\mathbf{u} \otimes (\mathbf{x} \mp |\mathbf{x}| \mathbf{e}_1)}{2|\mathbf{x}|^2 \mp 2|\mathbf{x}|x_1} \right) \cdot \mathbf{x} \\ &= \mathbf{x} - \frac{2\mathbf{u} (|\mathbf{x}|^2 \mp |\mathbf{x}|x_1)}{2(|\mathbf{x}|^2 \mp |\mathbf{x}|x_1)} \\ &= \mathbf{x} - \mathbf{u} \\ &= \mathbf{x} - \mathbf{x} \pm |\mathbf{x}| \mathbf{e}_1 \\ &= \pm |\mathbf{x}| \mathbf{e}_1 \end{aligned} \quad (2.4.39)$$

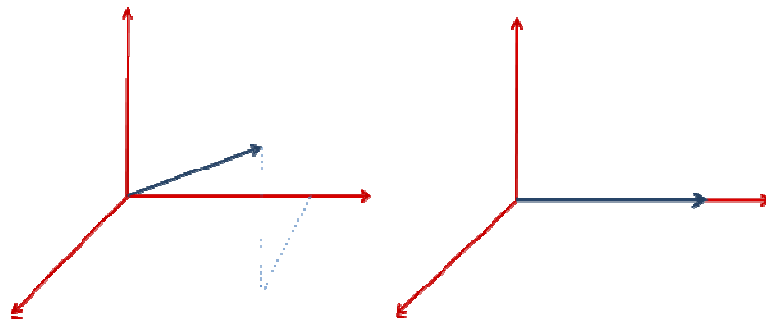


Figura 34: Efecto del operador de Householder sobre el vector  $\mathbf{x}$

Hay que recordar que P no es un operador de proyección, los operadores de proyección no son ortogonales. P es un operador de rotación, en una rotación la magnitud del vector no cambia, lo que si ocurre en un operador de proyección.

**2.4.7.3 Reducción de Householder.** El procedimiento para transformar una matriz **A** en tridiagonal es el siguiente:

El primer operador Householder P1, es seleccionado para rotar la sub-columna de la primera columna.

Que convierte  $\begin{bmatrix} a_{21} \\ a_{31} \\ \vdots \\ a_{n1} \end{bmatrix}$  en  $\begin{bmatrix} a'_{21} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$

El operador P1 tiene la siguiente forma:

$$P1 = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & & & & \\ 0 & & & & \\ \vdots & & & & \\ 0 & & & & \end{bmatrix} \begin{matrix} \\ \\ \\ (n-1)P_1 \\ \end{matrix} \quad (2.4.40)$$

Pre multiplicando (2.4.40) por (2.4.41)

$$P1 \cdot A = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & & & & \\ 0 & & & & \\ \vdots & & & & \\ 0 & & & & \end{bmatrix} \begin{matrix} \\ \\ \\ (n-1)P_1 \\ \end{matrix} \cdot \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} \\ a_{31} \\ \vdots \\ a_{n1} \end{bmatrix} \quad (2.4.41)$$

La matriz resultante A' sería la siguiente:

$$A' = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a'_{21} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (2.4.42)$$

Se puede observar que el operador actúa sobre la primera columna, si se quiere actuar sobre la primera fila se debe multiplicar (2.4.43) por P nuevamente.



$$\mathbf{P1} \cdot \mathbf{A} \cdot \mathbf{P1} = \begin{bmatrix} a_{11} & a'_{12} & 0 & \dots & 0 \\ a'_{21} & & & & \\ 0 & & & & \\ \vdots & & & & \\ 0 & & & & \end{bmatrix}$$

El segundo operador de Householder P2 tiene la siguiente forma:

$$\mathbf{P2} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & & & \\ \vdots & \vdots & & & \\ 0 & 0 & & & \end{bmatrix} \quad (2.4.43)$$

Que convierte  $\begin{bmatrix} a_{32} \\ a_{42} \\ \vdots \\ a_{n2} \end{bmatrix}$  en  $\begin{bmatrix} a'_{32} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$

A continuación se describen las operaciones con más detalle

$$\mathbf{P} = \mathbf{1} - \frac{\mathbf{u} \otimes \mathbf{u}}{\mathbf{H}}$$

$$\mathbf{A} \cdot \mathbf{P} = \mathbf{A} \cdot \left( \mathbf{1} - \frac{\mathbf{u} \otimes \mathbf{u}}{\mathbf{H}} \right)$$

$$\mathbf{A} \cdot \mathbf{P} = \mathbf{A} - \frac{\mathbf{A} \cdot \mathbf{u} \otimes \mathbf{u}}{\mathbf{H}}$$

Introduciendo un nuevo vector  $\mathbf{p} = \frac{\mathbf{A} \cdot \mathbf{u}}{\mathbf{H}}$

$$\mathbf{A} \cdot \mathbf{P} = \mathbf{A} - \mathbf{p} \otimes \mathbf{u}$$

Así que

$$\begin{aligned} \mathbf{A} \cdot \mathbf{P} \cdot \mathbf{A} &= \left( \mathbf{1} - \frac{\mathbf{u} \otimes \mathbf{u}}{\mathbf{H}} \right) \cdot (\mathbf{A} - \mathbf{p} \otimes \mathbf{u}) \\ &= \mathbf{A} - \mathbf{u} \otimes \mathbf{p} - \mathbf{p} \otimes \mathbf{u} + \frac{1}{\mathbf{H}} \mathbf{u} \otimes \mathbf{u} \cdot \mathbf{p} \otimes \mathbf{u} \\ &= \mathbf{A} - \mathbf{u} \otimes \mathbf{p} - \mathbf{p} \otimes \mathbf{u} + \frac{\mathbf{u} \cdot \mathbf{p}}{\mathbf{H}} \mathbf{u} \otimes \mathbf{u} \\ &= \mathbf{A} - \mathbf{u} \otimes \mathbf{p} - \mathbf{p} \otimes \mathbf{u} + 2\mathbf{K} \mathbf{u} \otimes \mathbf{u} \end{aligned}$$

Donde  $\mathbf{K} = \frac{\mathbf{u} \cdot \mathbf{p}}{2\mathbf{H}}$

Introduciendo el vector  $q$ ,  $q = p - Ku$

$$P \cdot A \cdot P = A - q \otimes u - u \otimes q$$

El algoritmo sería el siguiente:

$$\sigma = a_{21}^2 + a_{31}^2 + a_{41}^2 + \dots + a_{n1}^2$$

$$|x| = \sqrt{\sigma}$$

$$u = x \cdot \frac{1}{|x|} e_1$$

$$H = \frac{u \cdot u}{2}$$

$$p = \frac{A \cdot u}{H}$$

$$K = \frac{u \cdot p}{2H}$$

$$q = p - Ku$$

$$A' = A - q \otimes u - u \otimes q$$

De esta forma en n-2 pasos, se llegará a una matriz tridiagonal.

$P_{(n-2)} \dots P_3 P_2 P_1 A$  será una matriz de Hessenber inferior y

$P_{(n-2)} \dots P_3 P_2 P_1 A P_1 P_2 P_3 \dots P_{(n-2)}$  será una matriz tridiagonal

Si se desea encontrar los Eigenectores de A, es necesario almacenar las rotaciones  $P_1 P_2 P_3 \dots P_{(n-2)}$

A la matriz tridiagonal resultante, se le aplican sucesivas rotaciones de Jacobi

$Q_1 Q_2 Q_3 \dots Q_m$  hasta lograr una matriz diagonal.

Los Eigenectores de A serán:  $T = P_1 P_2 P_3 \dots P_{(n-2)} Q_1 Q_2 Q_3 \dots Q_m$

## **2.5. ANÁLISIS DE COMPONENTES PRINCIPALES**

El análisis de los componentes principales o ACP es una técnica del análisis estadístico multivariante que se clasifica entre los métodos de reducción simplificación o reducción de la dimensionalidad. Tiene sus antecedentes en psicología, a través de las técnicas de regresión lineal iniciadas por Galton. Sin embargo, desde hace algunos años se ha visto lo útil de su aplicación en las ciencias físicas, la ingeniería, la economía, la educación, el reconocimiento de patrones, etc.

Específicamente, Pearson [44], presentó la primera propuesta de componentes principales. Sin embargo, el nombre de “componentes principales” y su primer desarrollo teórico se debe a Hotteling [45] quien desarrolló el método de extracción de factores.

Por su parte, Thurstone [46], expresó la relación entre las correlaciones y las saturaciones de las variables en los factores. Asimismo, indujo el concepto de estructura simple y desarrollo la teoría y método de las rotaciones factoriales para obtener la estructura factorial más sencilla. En un principio las rotaciones eran graficas.

Kaiser [47], desarrolló el método llamado VARIMAX para realizar rotaciones ortogonales mediante procedimientos matemáticos.

A Fukuyama [48], se le considera como el primer investigador en aplicar ACP al reconocimiento de Patrones.

### **2.5.1. Ejemplo inicial**

Cuando se estudia un fenómeno que no se conoce, se toman medidas de variables como voltajes, velocidades, torque, etc. Desafortunadamente no se puede ver lo que está pasando, porque los datos aparecen en una nube de puntos; y gráficamente es difícil interpretar las relaciones entre las variables. Otra de las dificultades que se podría encontrar es que los datos presenten ruido y redundancia, estos son los principales problemas que enfrentan las ciencias empíricas y se presentan en sistemas complejos como la meteorología, neurociencias, fotometría y oceanografía.

Tomando un pequeño ejemplo de la física, mostrado en la figura 35. Se pretende estudiar el movimiento de un resorte ideal. El sistema consiste en una bola de masa  $m$  sujeta a un resorte de constante elasticidad  $k$  (la masa del resorte es despreciable). La bola es llevada a una distancia  $l$  de su posición de equilibrio. Ya que el resorte es ideal debería oscilar indefinidamente a través del eje X.

Este es un problema estándar de la física, en el cual el movimiento alrededor del eje X, puede ser resuelto a través de una función del tiempo, es decir de una sola variable X.

Sin embargo si no se entiende el fenómeno, no se conocería nada de esto; no se conoce cuales son las dimensiones importantes para medir; así que se decide medir la posición de la bola en un espacio tridimensional. Para medir la posición, se dispone de tres videocámaras alrededor del sistema (ver figura 35).

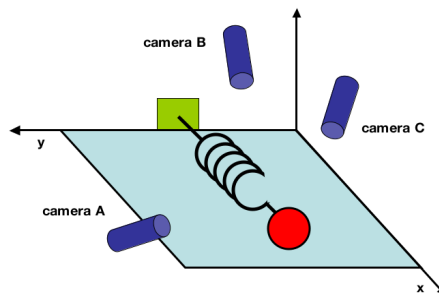


Figura 35: Ejemplo de una masa sujeta a un resorte.

Cada cámara contribuye a una proyección bidimensional de la posición de la bola, de esta forma la posición de la bola quedará descrita por el vector X.

$$X = \begin{bmatrix} x_A \\ y_A \\ x_B \\ y_B \\ x_C \\ y_C \end{bmatrix} \quad (2.5.1)$$

Donde  $x_A, y_A$  son las proyecciones de la posición de la bola en X e Y según la cámara A, y así sucesivamente.

Si se toman 10 medidas de posición de la bola, los datos podrían dar algo como esto:

	xA	yA	xB	yB	xZ	yZ
1	6,15	9,31	6,13	9,32	6,11	9,30
2	6,33	9,31	6,32	9,32	6,34	9,30
3	6,95	9,31	6,95	9,32	6,93	9,30
4	7,57	9,31	7,53	9,32	7,56	9,30
5	8,39	9,31	8,33	9,32	8,37	9,30
6	11,35	9,31	11,38	9,32	11,36	9,30
7	12,22	9,31	12,25	9,32	12,25	9,30
8	12,69	9,31	12,63	9,32	12,64	9,30
9	12,86	9,31	12,89	9,32	12,83	9,30
10	12,93	9,31	12,94	9,32	12,93	9,30

Tabla 2: Ejemplo de los datos tomados por la cámara.

Notar que las proyecciones en X:  $x_A, x_B, x_Z$  son semejantes; indicando que todas las cámaras detectan el movimiento en X con algún grado de error. Las Proyecciones en Y:  $y_A, y_B, y_Z$  son casi iguales y constantes indicando que el movimiento no tiene desplazamiento en este eje.

En todas las medidas se introdujeron errores.

No se puede representar geoméricamente los datos, ya que, este es un espacio de seis dimensiones, solo se puede representar gráficamente hasta tres dimensiones. Si se representa en una grafica, únicamente los valores de  $x_A, x_B, x_Z$ . (Ver figura 36).

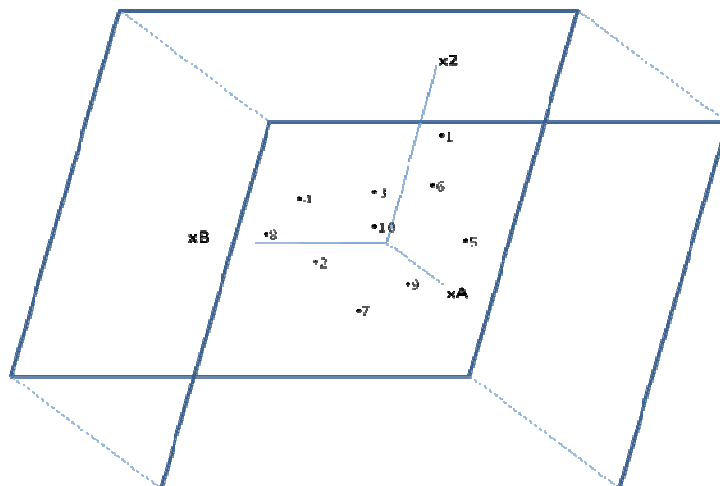


Figura 36: Vista de los datos en 3 dimensiones.

En este momento no es posible realizar una interpretación, porque no se posee una perspectiva clara de los datos. Pero, si se rotan los datos, se tendría lo siguiente (ver figura 37):

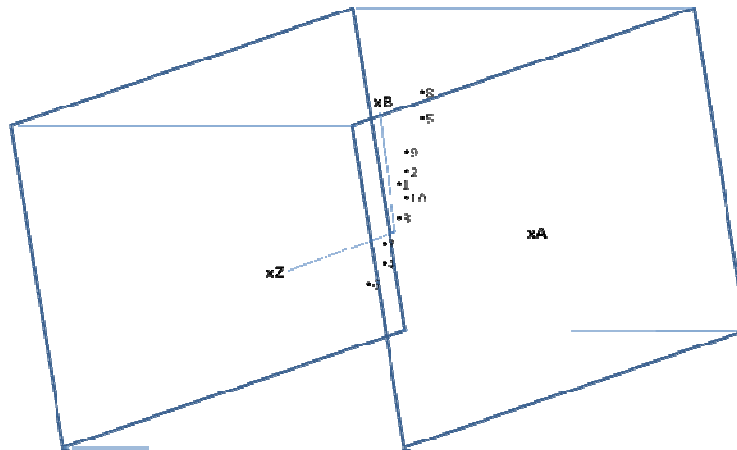


Figura 37: Rotación de la vista de los datos.

Se puede ver que los datos están prácticamente sobre un plano. Esto pone en manifiesto que no es necesario tres dimensiones para representar los datos, sino, solamente dos.

Se tiene entonces, que buscar un sistema de referencia para el plano (subespacio) más cercano a la nube de puntos de forma que, al proyectarlos todos sobre dicho plano, la pérdida de información sea mínima.

La pérdida de información puede entenderse en términos de variabilidad del conjunto de puntos (varianza) o en términos de la similitud entre las interdistancias entre los puntos, calculadas en el espacio original y las calculadas en la proyección del subespacio.

El subespacio quedará definido mediante un sistema de referencia para el mismo, es decir, mediante dos vectores perpendiculares dentro del subespacio. El primero se sitúa en la dirección en la que más varían los datos, el segundo, perpendicular al primero recogiendo la mayor parte de la variabilidad restante y así sucesivamente.

Los vectores del sistema de referencia definen nuevas variables, que son combinaciones lineales de las variables de partida y se denominan componentes principales. De esta forma, se puede reducir la dimensión seleccionando solamente las primeras componentes.

La reducción de la dimensión se deriva del hecho de que las variables están relacionadas entre sí y, por tanto, tienen información común, de alguna manera, la información común a todas ellas se extrae en las componentes principales.

La siguiente figura (Figura 38) (plano sobre el que están los datos del espacio de tres dimensiones), muestra lo anterior geoméricamente.

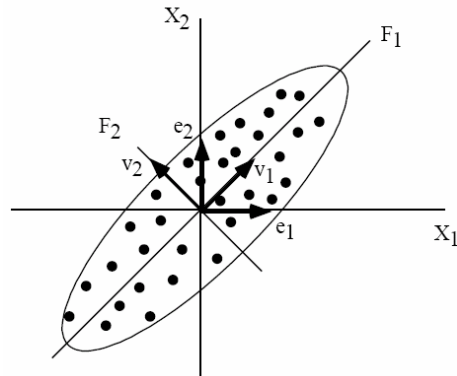


Figura 38: Nuevo sistema de referencia para los datos del resorte.

Donde  $e_1, e_2$  son las base originales, los datos son encerrados en una elipse, donde la parte más ancha indica mayor variabilidad y por lo tanto la dirección de la nueva base  $v_1$ , la siguiente base  $v_2$  recoge lo que queda de la variabilidad.

**2.5.3. Modelo matemático.** Si cada individuo (dato) esta descrito por  $k$ -variables. Los datos de  $n$  sujetos se pueden representar mediante la matriz  $X$  de orden  $n \times k$

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1k} \\ x_{21} & x_{22} & \dots & x_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nk} \end{bmatrix} \quad (2.5.2)$$

Los datos de la matriz forman una nube de  $n$  puntos en un espacio  $k$ -dimensional.

Por lo tanto sea  $p_1, p_2, \dots, p_k$  una base de  $R^k$ , y considerando la nube  $k$ -dimensional de  $n$  puntos a la que se le denominará  $M_1, M_2, \dots, M_n$ , los cuales tendrá coordenadas  $(x_{i1}, x_{i2}, \dots, x_{ik})$  en la base  $p_1, p_2, \dots, p_k$ .

Por lo tanto, cada fila en la matriz  $X$  definida en (2.5.2) corresponde a los componentes de un punto de la nube en el sistema de coordenadas introducido por la base.

Considerando ahora otra base  $p_1, p_2, \dots, p_k$  de  $R^k$ . y las coordenadas  $(z_{i1}, z_{i2}, \dots, z_{ik})$  del punto  $M_i$  en esta nueva base, se puede construir la matriz  $Z$  de orden  $n \times k$ .

$$Z = \begin{bmatrix} z_{11} & z_{12} & \dots & z_{1k} \\ z_{21} & z_{22} & \dots & z_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ z_{n1} & z_{n2} & \dots & z_{nk} \end{bmatrix} \quad (2.5.3)$$

Para relacionar  $Z$  con  $X$ , se construye  $U$ , como una matriz de paso de la base  $p_1, p_2, \dots, p_k$  a la nueva base  $p'_1, p'_2, \dots, p'_k$ . La matriz  $U$  es una matriz de orden  $k \times k$  cuyas columnas tienen las coordenadas de los vectores  $p'_1, p'_2, \dots, p'_k$  en la base inicial, es decir, si

$$\begin{aligned} p'_1 &= u_{11}p_1 + u_{21}p_2 + \dots + u_{k1}p_k \\ p'_2 &= u_{12}p_1 + u_{22}p_2 + \dots + u_{k2}p_k \\ &\dots \\ p'_k &= u_{1k}p_1 + u_{2k}p_2 + \dots + u_{kk}p_k \end{aligned} \quad (2.5.4)$$

Entonces la matriz de paso  $U$  será:

$$U = \begin{bmatrix} u_{11} & u_{12} & \dots & u_{1k} \\ u_{21} & u_{22} & \dots & u_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ u_{k1} & u_{k2} & \dots & u_{kk} \end{bmatrix} \quad (2.5.5)$$

La relación entre  $Z$ ,  $X$  y  $U$  será:

$$X = Z U^T \quad (2.5.6)$$

Como  $U$  es invertible se puede deducir que:

$$X (U^T)^{-1} = Z \quad (2.5.7)$$

Si los vectores  $p_1, p_2, \dots, p_k$  forman una base orto normal, la matriz  $U$  satisface la relación:

$$U^T U = I \quad (2.5.8)$$

Y por lo tanto:

$$(U^T)^{-1} = U \quad (2.5.9)$$

Se tienen las siguientes interpretaciones de  $U$ :



- \* U es una matriz que transforma X en Z.
- \* Geométricamente U es una rotación.

**2.5.2.1 Matriz de covarianza.** La matriz de covarianza para la matriz X será.

$$S_x = \begin{bmatrix} \sigma^2_{x1} & \sigma_{x1x2} & \dots & \sigma_{x1xk} \\ \sigma_{x2x1} & \sigma^2_{x2} & \dots & \sigma_{x2xk} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{kx1} & \sigma_{kx2} & \dots & \sigma^2_{xk} \end{bmatrix} \quad (2.5.10)$$

Donde  $\sigma^2_{x_i}$  representa la varianza de la variable  $X_i$  en el conjunto y  $\sigma_{ij}$  representa la covarianza de  $X_i$  y  $X_j$ .

Para simplificar, se asume que las variables  $X_1, X_2, \dots, X_k$  están centradas (tienen media cero), la matriz de covarianza  $S_x$  se puede expresar como:

$$S_x = \frac{1}{n} X^T X \quad (2.5.11)$$

La matriz  $S_x$  es una matriz cuadrada simétrica de orden  $k \times k$ .

La matriz  $S_x$  captura todas las posibles interrelaciones entre todos los posibles pares de variables a medir. Los valores de la matriz reflejan el ruido y redundancia en la medida de las variables

**2.5.2.2. Diagonalización de la Matriz de covarianza.** Como se vio en el numeral 2.3, conceptos de estadística, la covarianza representa la redundancia, y la varianza el nivel de señal-ruido, por lo que tienen dos objetivos principales:

- \*) Minimizar la redundancia, medida a través de la covarianza.
- \*) Maximizar la señal medida a través de la varianza.

Por definición de covarianza, esta es no negativa, así que el valor mínimo de covarianza es cero. Qué forma debería tener la matriz de covarianza de Z ( $S_z$ )? evidentemente, debería ser una matriz donde todos los términos de que no se encuentren en la diagonal sean ceros. La matriz de covarianza de Z ( $S_z$ ), debe ser diagonal.

La matriz de covarianza de Z es igual, a

$$S_z = \frac{1}{n} Z^T Z \quad (2.5.12)$$

De (2.5.7)  $Z = X (U^T)^{-1}$

$$S_z = \frac{1}{n} (X (U^T)^{-1})^T (X (U^T)^{-1})$$

$$S_z = \frac{1}{n} U^{-1} X^T X (U^T)^{-1}$$

$$S_z = U^T S_x U \tag{2.5.13}$$

Esta última ecuación indica que la matriz de covarianza de Z,  $S_z$ , puede ser calculada en términos de la matriz de covarianza de X,  $S_x$ .

Una matriz cuadrada simétrica es diagonalizada por otra matriz que contenga los eigenvectores de esta [49], así que, si se escoge U como una matriz de eigenvectores de  $S_x$ , diagonaliza  $S_x$ . que es lo que se busca.

Para diagonalizar  $S_x$  (encontrar los eigenvectores) se procede como se vió en el numeral 2.4.7.

## 2.6. CONCEPTOS DE IMAGENES DIGITALES

**2.6.1. Pixel.** Una computadora representa los colores como una combinación de diferentes intensidades de rojo, verde y azul.

El color ■ (verde claro) es representado como rojo =128, verde=255, azul = 0.

**2.6.2. Imagen Digital.** Una imagen  $I(x,y)$  se define como un arreglo bidimensional de  $n \times m = N$  pixeles.

Para el proyecto se trabajó con imágenes de  $92 \times 112 = 10304$  pixeles. Se dice digital porque los colores son representados como números que pueden ser accedidos a través de los índices  $x,y$ . (Ver figura 39).



Figura 39: Imagen Digital.

**2.6.3. Escala de grises.** Una imagen esta en escala de grises si cada pixel puede tener 256 (8 bits) valores diferentes de intensidad de gris.

La escala de grises se conforma con una combinación de igual intensidad de rojo, verde, azul, por ejemplo:

El color ■ (nivel de gris 110) es igual a: rojo=110, verde=110, azul = 110.

El color ■ (nivel de gris 200) es igual a: rojo =200, verde=200, azul = 200.

En el proyecto las imágenes son almacenadas en escala de grises.

Para convertir un color (R,G,B) a un nivel de gris, se puede realizar por una función de conversión

$\text{Gray}(\text{Pixel}) = a \cdot R + b \cdot G + c \cdot B$ , donde  $a+b+c = 1$ . El ejemplo más simple  
 $\text{Gray}(\text{Pixel}) = 0.33 \cdot R + 0.33 \cdot G + 0.33 \cdot B$

Un ejemplo de conversión, se muestra en la siguiente figura. (Figura 40).

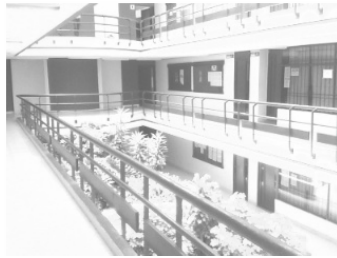


Figura 40: imagen en escala de grises.

**2.6.3.1. Imagen binaria.** Es un caso particular de imagen en escala de grises donde los niveles de intensidad de gris son 0 o 255, es decir negro o blanco (ver figura 41).

Para binarizar una imagen, previamente se convierte a escala de grises. Los pixeles de la imagen binaria se obtienen por comparación de un umbral:

0 si  $\text{Gray}(\text{Pixel}) < \text{umbral}$   
255 si  $\text{Gray}(\text{Pixel}) > \text{umbral}$

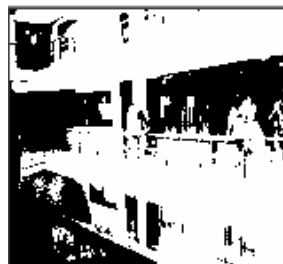


Figura 41: imagen binaria

**2.6.4. Imagen como un vector.** Una imagen puede ser representada por un vector de dimensión  $N = n \times m$ , concatenando las filas consecutivamente (ver figura 42), de esta forma una imagen se puede ver como un punto en un espacio de  $N$  dimensiones.



Figura 42: Imagen como un vector.

**2.6.5. Filtros.** En el proyecto se implemento un filtro de media, el cual se describe de la siguiente manera, se toman una subregión de 3x3 de la imagen, se suman sus elementos y se divide sobre 9, el resultado (promedio) es asignado a cada pixel de la región.

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

$$\text{suma} = (a+b+c+d+e+f+g+h+i)/9$$

$$\begin{bmatrix} \text{suma} & \text{suma} & \text{suma} \\ \text{suma} & \text{suma} & \text{suma} \\ \text{suma} & \text{suma} & \text{suma} \end{bmatrix}$$

En la figura 43, se ve el resultado de aplicar el filtro.

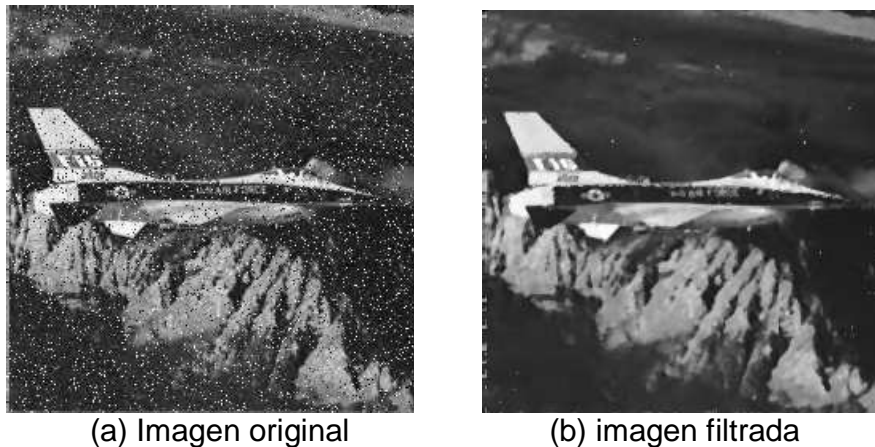


Figura 43: Proceso de filtrado de una imagen.

**2.6.6. Segmentación de imágenes.** La segmentación tiene como objetivo quitar de la imagen original información que no es relevante.

En el proyecto se toma como suposición que las imágenes para entrenamiento como para reconocimiento son tomadas en un fondo blanco, aunque funciona en un fondo moderadamente complejo.

Con la anterior suposición, para una persona es fácil indicar cuál es el comienzo y fin del rostro en filas y columnas.

Para la segmentación se tiene en cuenta que cuando hay un rostro presente en la imagen, este cambia bruscamente los pixeles de fondo, por lo que se puede

modelar este cambio como una derivada de la cantidad de elementos en filas y columnas.

El primer paso consiste en adquirir la imagen, luego es convertida a escala de grises y se aplican filtros, después es convertida a imagen binaria para que los cambios se hagan más bruscos, se construye una función que suma los elementos negros (0 nivel de gris) de cada columna, y luego se calcula la derivada de esta función en dos intervalos: el primer intervalo comienza en el origen hasta la mitad de la imagen, el segundo intervalo comienza en la mitad de la imagen hasta el fin de esta. Se encuentra el índice donde la derivada sea máxima en el primer intervalo, y el índice donde la derivada es mínima en el segundo intervalo. Estos índices indicaran el origen y el fin del rostro. Un proceso similar se aplica para encontrar el comienzo de la imagen en Y. La figura 44, muestra el proceso.

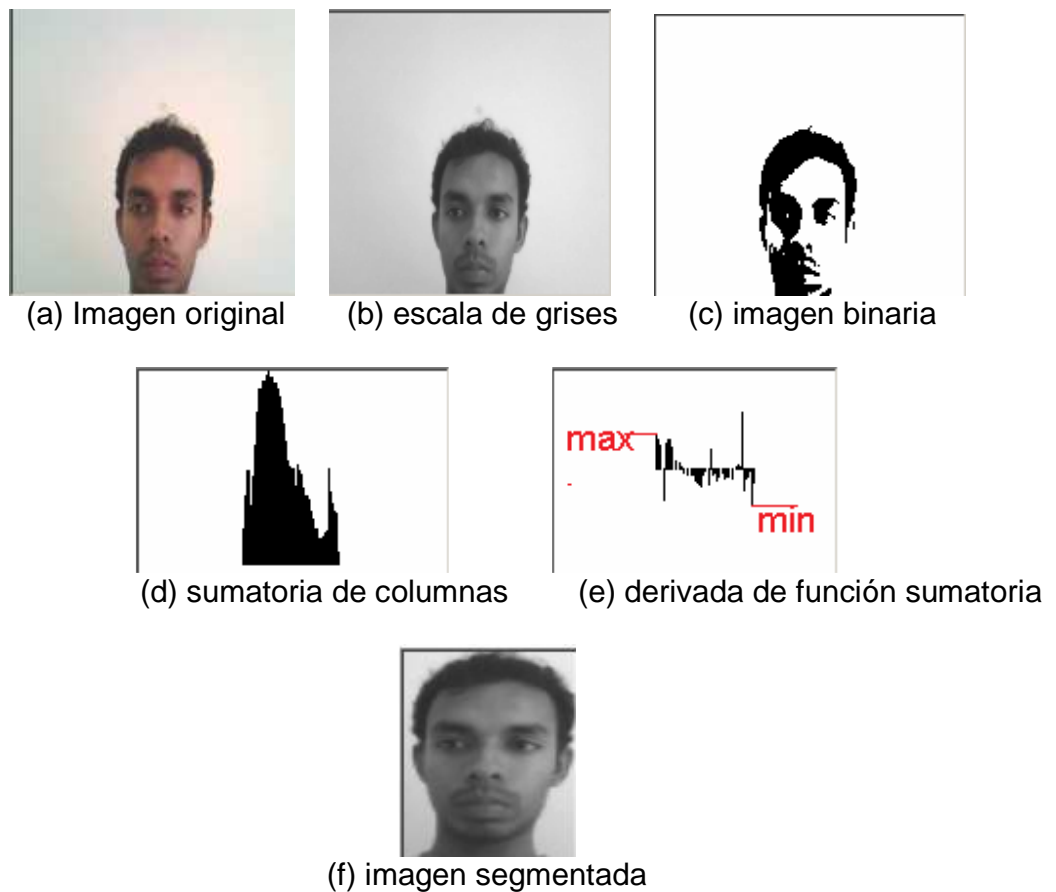


Figura 44: Proceso de segmentación de imagen.

## 2.7. ALGORITMO EIGENFACES

Inventado por Turk y Pentland [2], el algoritmo Eigenfaces utiliza la técnica de Análisis de Componentes Principales para describir las imágenes y realizar reconocimiento por medio de la proyección de las imágenes de entrenamiento en el FaceSpace, por lo que sus pasos serán muy similares a los ACP.

**2.7.1. Modelo matemático.** Dado el conjunto de muestras de imágenes de rostros en vectores columnas

$$\Gamma_0, \Gamma_1, \Gamma_2, \dots, \Gamma_{M-1}$$

Se define como imagen media de las muestras al vector columna

$$\Psi = 1/M \sum_{j=0}^{M-1} \Gamma_j \quad (2.7.1)$$

Se definen los siguientes nuevos vectores con media nula

$$\Phi_i = \Gamma_i - \Psi \quad (2.7.2)$$

La matriz de covarianza se define como

$$C = 1/M \sum_{j=0}^{M-1} \Phi_j \Phi_j^T \quad (2.7.3)$$

Como se vió del análisis de componentes principales, el objetivo es encontrar los vectores que mejor representen los datos, en este caso, las imágenes. Estos vectores definen un subespacio correspondiente a las imágenes de los rostros y es llamado entonces, face subspace. Estos vectores son los eigenvectores de la matriz de covarianza de las imágenes originales; y por tal razón, son llamados Eigenfaces.

Teóricamente la matriz C tiene un total de N autovectores y N autovalores, lo cual representa un problema desde el punto de vista computacional ya que, el rango de la matriz C es de 10 304 x 10304. Por otro lado el numero de muestras es bastante inferior al espacio de las imágenes, es decir  $M \ll N$ , por lo tanto, existirán solo M autovectores que concentraran la información del subespacio de las muestras.

M. Turk y A. Pentland presenta un método para reducir el problema de dimensiones  $N \times N$  a un problema de dimensiones  $M \times M$ , lo cual lo hace realizable

desde punto de vista computacional. Por ejemplo, en caso de tener 16 muestras de imágenes, el problema se reduciría de una dimensión de 10 304 x 10304 a 16 x 16.

Se define la matriz A de NxM (10 304 x 16 por ejemplo) cuyas columnas son los vectores de las imágenes con promedio nulo, es decir

$$A = [\Phi_0 \Phi_1 \Phi_2 \dots \Phi_{M-1}] \quad (2.7.4)$$

Puede verse fácilmente que la ecuación (2.7.3) puede escribirse en términos de la matriz A de la siguiente forma

$$C = \frac{1}{M} \sum_{j=0}^{M-1} \Phi_j \Phi_j^T = \frac{1}{M} A A^T \quad (2.7.5)$$

Si se considera la diagonalización de la matriz  $A^T A$  de  $M \times M$ , lo cual implica resolver la siguiente ecuación

$$A^T A v_i = \mu_i v_i \quad (2.7.6)$$

Con  $i = 0, 1, 2, \dots, M-1$  y donde los  $v_i$  son los eigenvectores y los  $\mu_i$  son los eigenvalores de la matriz  $A^T A$

Si se premultiplica la ecuación (2.7.6) por  $\frac{1}{M} A$ , se obtiene

$$\frac{1}{M} A A^T (A v_i) = \frac{1}{M} \mu_i (A v_i) \quad (2.7.7)$$

Y usando la ecuación (2.7.5) se tiene

$$C (A v_i) = 1/M \mu_i (A v_i) \quad (2.7.8)$$

Es decir que los autovalores y autovectores de la matriz C se puede obtener utilizando los autovalores y autovectores de la matriz reducida  $A^T A$  y las ecuaciones, son

$$u_i = A v_i \quad (2.7.9)$$

$$\lambda_i = \frac{1}{M} \mu_i \quad (2.7.10)$$

De acuerdo con la teoría de Análisis de Componentes Principales, cualquier imagen de las muestras de aprendizaje tiene una representación exacta a través de los coeficientes de proyección sobre el subespacio. Además, se puede demostrar que los autovalores de mayor valor son los que tienen asociados las Eigenfaces que contienen mayor información acerca de la imagen representada.



Según la teoría del Análisis de los Componentes Principales una imagen del conjunto de muestras se puede proyectar en Face space según la ecuación

$$\Gamma_j' = U^T (\Gamma_j - \Psi); \quad j=0,1,2,\dots,M-1 \quad (2.7.11)$$

**2.7.2. Proceso de reconocimiento.** Si se desea reconocer una imagen de prueba  $\Gamma$ , se proyecta esta imagen en el FaceSpace.

$$\Omega = U^T(\Gamma - \Psi) \quad (2.7.12)$$

Se Calcula la distancia euclídea de la imagen de prueba a cada una de las imágenes de la muestra

$$\epsilon_j^2 = \|\Omega - \Omega_j\|^2; \quad j = 0,1,2,\dots,M-1 \quad (2.7.13)$$

Calculamos el umbral de distancia definido como un medio de máxima distancia entre dos imágenes del conjunto de todas las imágenes

$$\Theta_c = \frac{1}{2} \max_{i,j} \{\|\Omega_i - \Omega_j\|\}; \quad i,j = 0,1,2,\dots,M-1 \quad (2.7.14)$$

Se calcula la distancia,  $\epsilon$ , entre la imagen original  $\Gamma$  y su imagen reconstruida del espacio del EigenFace,  $\Gamma_f$

$$\epsilon^2 = \|\Gamma - \Gamma_f\|^2, \text{ donde } \Gamma_f = U * \Omega + \Psi \quad (2.7.15)$$

- Si  $\epsilon \geq \Theta_c$   
Entonces la imagen de prueba no es una imagen de un rostro.
- si  $\epsilon < \Theta_c$  y  $\epsilon_j \geq \Theta_c$  para todas las imágenes de muestra,  
Entonces la imagen de prueba contiene una imagen conocida.
- Si  $\epsilon < \Theta_c$  y  $\epsilon_j^* = \min_j \{\epsilon_j\} < \Theta_c$ ,  
Entonces la imagen de prueba pertenece al rostro del individuo  $j$

**2.7.3. Resumen gráfico del algoritmo.** Para el siguiente ejemplo se toma el umbral de error  $\Theta_c = 9.32$

Sean siguientes son imágenes de entrenamiento, figura 45:



Figura 45: Ejemplo imagenes de entrenamiento.

Se concatena cada imagen  $\Gamma_0, \Gamma_1, \Gamma_2$ . Si se supone que la imagen  $I_0(x,y)$  es representada por la siguiente matriz.

$$\begin{bmatrix} 1 & 2 & 3 \\ 3 & -1 & 2 \\ 4 & 5 & 1 \end{bmatrix}_{3 \times 3} \quad \text{Es transformado al vector columna } \Gamma_0 = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 3 \\ -1 \\ 2 \\ 2 \\ 4 \\ 5 \\ 5 \\ 1 \\ 1 \end{bmatrix}_{9 \times 1}$$

Y de igual manera con todas las imágenes.

Se calcula la media de todas las imágenes  $\Psi$ , a través de la fórmula (2.7.1).

$$\begin{array}{cccc} \Gamma_1 & \Gamma_2 & \Gamma_3 & \Psi = (\Gamma_1 + \Gamma_2 + \Gamma_3) / 3 \\ \begin{bmatrix} 1 \\ 2 \\ 0 \\ 3 \\ -1 \\ 2 \\ 4 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 2 \\ 3 \\ 4 \\ 1 \\ 6 \\ 8 \\ 1 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \\ 5 \\ 2 \\ -2 \\ -1 \\ 1 \\ 0 \\ 3 \end{bmatrix} & \begin{bmatrix} 1 \\ 2 \\ 3 \\ 2 \\ 1 \\ 3 \\ 2 \\ 0 \\ 1 \end{bmatrix} \end{array}$$

Se resta a cada imagen la media  $\Phi_i = \Gamma_i - \Psi$

$$\begin{array}{ccc} \Phi_i = \Gamma_i - \Psi & & \\ \begin{bmatrix} 0 \\ 0 \\ -3 \\ 1 \\ -2 \\ -1 \\ 2 \\ 0 \\ -1 \end{bmatrix} & \begin{bmatrix} 1 \\ 1 \\ -2 \\ -1 \\ 5 \\ 5 \\ -1 \\ 0 \\ -1 \end{bmatrix} & \begin{bmatrix} -1 \\ -1 \\ 2 \\ 0 \\ -3 \\ -2 \\ -1 \\ 0 \\ 2 \end{bmatrix} \\ \Phi_1 & \Phi_2 & \Phi_3 \end{array}$$

Para calcular la matriz de covarianza, primero se multiplica  $A^T A$

$$A = \begin{bmatrix} 0 & 1 & -1 \\ 0 & 1 & -1 \\ -3 & -2 & 2 \\ 1 & -1 & 0 \\ -2 & 5 & -3 \\ -1 & 5 & -2 \\ 2 & -1 & -1 \\ 0 & 0 & 0 \\ -1 & -1 & 2 \end{bmatrix} \quad A^T = \begin{bmatrix} 0 & 0 & -3 & 1 & -2 & -1 & 2 & 0 & 1 \\ 1 & 1 & -2 & -1 & 5 & 5 & -1 & 0 & -1 \\ -1 & -1 & 2 & 0 & -3 & -2 & -1 & 0 & 2 \end{bmatrix}$$

$\Phi_1 \quad \Phi_2 \quad \Phi_3$

$$A^T A = \begin{bmatrix} 20 & -11 & -2 \\ -11 & 59 & -32 \\ -2 & -32 & 24 \end{bmatrix}$$

Se calcula los eigenvectores de la matriz anterior:  $v_1, v_2, v_3$

$$\begin{bmatrix} 0.3923 \\ 0.5060 \\ 0.7681 \end{bmatrix} \quad \begin{bmatrix} 0.9087 \\ -0.0835 \\ -0.4091 \end{bmatrix} \quad \begin{bmatrix} 0.1429 \\ -0.8585 \\ 0.4926 \end{bmatrix}$$

$v_1 \quad v_2 \quad v_3$

Se calcula los eigenvectores de la matriz de covarianza por la formula (2.7.9)

$$u_i = A v_i$$

$$\begin{bmatrix} -0.2621 \\ -0.2621 \\ -0.6527 \\ -0.1137 \\ -0.5589 \\ 0.6015 \\ -0.4895 \\ 0 \\ 0.6379 \end{bmatrix} \quad \begin{bmatrix} 0.3256 \\ 0.3256 \\ -3.3773 \\ 0.9922 \\ -1.0076 \\ -0.5080 \\ 2.3100 \\ 0 \\ -1.6434 \end{bmatrix} \quad \begin{bmatrix} -1.3511 \\ -1.3511 \\ 2.2735 \\ 1.0014 \\ -6.0561 \\ -5.4206 \\ 0.6517 \\ 0 \\ 1.7008 \end{bmatrix}$$

$u_1 \quad u_2 \quad u_3$

Se Forma la matriz U con los eigenvectores anteriores al cual se llama Eigenfaces

$$U = \begin{bmatrix} -0.2621 & 0.3256 & -1.3511 \\ -0.2621 & 0.3256 & -1.3511 \\ -0.6527 & -3.3773 & 2.2735 \\ -0.1137 & 0.9922 & 1.0014 \\ -0.5589 & -1.0076 & -6.0561 \\ 0.6015 & -0.5080 & -5.4206 \\ -0.4895 & 2.3100 & 0.6517 \\ 0 & 0 & 0 \\ 0.6379 & -1.6434 & 1.7008 \end{bmatrix}$$

$u_1 \quad u_2 \quad u_3$

Los Eigenfaces se puede representar como imágenes (ver figura 46), ya que tienen la misma dimension que las imágenes de entrenamiento:



Figura 46: Representación grafica de los Eigenfaces.

Se proyecta las imágenes originales en el FaceSpace

$$U^T = \begin{bmatrix} -0.2621 & -0.2621 & -0.6527 & -0.1137 & -0.5589 & 0.6015 & -0.4895 & 0 & 0.6379 \\ 0.3256 & 0.3256 & -3.3773 & 0.9922 & -1.0076 & -0.5080 & 2.3100 & 0 & -1.6434 \\ -1.3511 & -1.3511 & 2.2735 & 1.0014 & -6.0561 & -5.4206 & 0.6517 & 0 & 1.7008 \end{bmatrix}$$

$$\Phi_1 = \begin{bmatrix} 0 \\ 0 \\ -3 \\ 1 \\ -2 \\ -1 \\ 2 \\ 0 \\ -1 \end{bmatrix} \quad \Omega_1 = U^T \Phi_1 = \begin{bmatrix} 0.7438 \\ 19.9107 \\ 11.3163 \end{bmatrix}$$

$$\Phi_1 = \begin{bmatrix} 0 \\ 0 \\ -3 \\ 1 \\ -2 \\ -1 \\ 2 \\ 0 \\ -1 \end{bmatrix} = \text{[Image of a man's face]} = 0.73^* \text{[Image of a man's face]} + 19.91^* \text{[Image of a woman's face]} + 11.31^* \text{[Image of a man's face]}$$

Figura 47: Imagen como una combinación de pesos en FaceSpace.

Lo que da un vector de peso por cada imagen en la base de entrenamiento

Si se supone que se quiere reconocer la siguiente persona, figura 48:



Figura 48: Imagen de prueba para ser reconocida.

Se convierte a vector y se resta la media de las imágenes, según la ecuación (2.7.2) resultando el siguiente vector:

$$\Phi = \begin{bmatrix} 5 \\ -2 \\ 1 \\ 0 \\ 4 \\ -3 \\ 0 \\ 2 \\ -1 \end{bmatrix}$$

Se proyecta el vector anterior en el FaceSpace obteniéndose el vector de pesos:

$$\Omega = U^T \Phi = \begin{bmatrix} 3.1137 \\ 43.10 \\ 8.61 \end{bmatrix}$$

La distancia de este vector con la de la primera persona es  $d=23.46$ , ya que no es menor que  $\Theta_c$  (definido al principio del ejemplo), se dice que la imagen de prueba no pertenece a la primera persona, y así con el resto. Muy seguramente ninguna cumpla con la condición, por lo tanto, no se reconoce la persona.

### **3. METODOLOGIA Y HERRAMIENTAS PARA EL DESARROLLO DE FACEMATH**

#### **3.1. RUP: PROCESO RACIONAL UNIFICADO.**

Los orígenes de RUP se remontan al modelo espiral, es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. RUP es en realidad un refinamiento realizado por Rational Software del más genérico Proceso Unificado. Sus principales características son [50]:

Forma disciplinada de asignar tareas y responsabilidades (quien hace que, cuando y como).

Pretende implementar las mejores prácticas en Ingeniería de Software

- Desarrollo iterativo
- Administración de requisitos
- Uso de arquitectura basada en componentes
- Control de cambios
- Modelado visual del software
- Verificación de la calidad del software

El RUP es un producto de Rational (IBM), se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso, incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso).

El RUP divide el proceso de desarrollo en ciclos, teniendo un producto final al final de cada ciclo, cada ciclo se divide en fases que finalizan con un hito donde se debe tomar una decisión importante:

**Inicio:** se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos.

**Elaboración:** se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos.

**Construcción:** se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario.

Transición: se implementa el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto suelen surgir nuevos requisitos a ser analizados.

Fases del RUP:

- Establece oportunidad y alcance
- Identifica las entidades externas o actores con las que se trata
- Identifica los casos de uso.

## 3.2. .NET FRAMEWORK

**3.2.1. Información general y conceptual sobre .NET Framework.** .NET Framework es un componente integral de Windows que admite la creación y la ejecución de la siguiente generación de aplicaciones y servicios Web XML. El diseño de .NET Framework está enfocado a cumplir los objetivos siguientes:

- Proporcionar un entorno coherente de programación orientada a objetos, en el que el código de los objetos se pueda almacenar y ejecutar de forma local, ejecutar de forma local pero distribuida en Internet o ejecutar de forma remota.
- Proporcionar un entorno de ejecución de código que reduzca lo máximo posible la implementación de software y los conflictos de versiones.
- Ofrecer un entorno de ejecución de código que fomente la ejecución segura del mismo, incluso del creado por terceras personas desconocidas o que no son de plena confianza.
- Proporcionar un entorno de ejecución de código que elimine los problemas de rendimiento de los entornos en los que se utilizan secuencias de comandos o intérpretes de comandos.
- Ofrecer al programador una experiencia coherente entre tipos de aplicaciones muy diferentes, como las basadas en Windows o en el Web.
- Basar toda la comunicación en estándares del sector para asegurar que el código de .NET Framework se puede integrar con otros tipos de código.

.NET Framework contiene dos componentes principales:

**Common Language Runtime:** es el fundamento de .NET Framework. El motor en tiempo de ejecución se puede considerar como un agente que administra el código en tiempo de ejecución y proporciona servicios centrales, como la administración de memoria, la administración de subprocesos y la interacción remota, al tiempo que aplica una seguridad estricta a los tipos y otras formas de especificación del código que fomentan su seguridad y solidez.

**Biblioteca de clases de .NET Framework:** Es una completa colección orientada a objetos de tipos reutilizables que se pueden emplear para desarrollar aplicaciones que abarcan desde las tradicionales herramientas de interfaz gráfica de usuario (GUI) o de línea de comandos hasta las aplicaciones basadas en las innovaciones más recientes proporcionadas por ASP.NET, como los formularios Web Forms y los servicios Web XML.

### **3.3. WPF WINDOWS PRESENTATION FOUNDATION**

**3.3.1. Introducción.** Windows Presentation Foundation (WPF) es un sistema de presentación de la próxima generación, para crear aplicaciones cliente de Windows que proporcionen una experiencia impactante para el usuario desde el punto de vista visual. Con WPF, puede crear una amplia gama de aplicaciones independientes y hospedadas en explorador.

El núcleo de WPF es un motor de representación independiente de la resolución y basado en vectores construido para aprovechar al máximo el hardware de gráficos moderno. WPF amplía el núcleo con un completo conjunto de características de programación de aplicaciones, entre las que se incluyen Lenguaje de marcado de aplicaciones extensible (XAML), controles, enlace de datos, diseño, gráficos 2D y 3D, animación, estilos, plantillas, documentos, multimedia, texto y tipografía. WPF se incluye en Microsoft .NET Framework, lo que permite crear aplicaciones que incorporen otros elementos de la biblioteca de clases de .NET Framework.

**3.3.2. XAML.** En general, se utiliza el lenguaje marcado de aplicaciones extensible (XAML) para implementar la apariencia de una aplicación, y los lenguajes de programación administrados (subyacentes) para implementar su comportamiento. Esta separación entre la apariencia y el comportamiento aporta las ventajas siguientes:

- Se reducen los costos de programación y mantenimiento, al no estar el marcado específico de la apariencia estrechamente relacionado con el código específico del comportamiento.
- La programación es más eficaz porque los diseñadores pueden implementar la apariencia de una aplicación al mismo tiempo que los programadores implementan su comportamiento.
- Se pueden utilizar varias herramientas de diseño para implementar y compartir el lenguaje marcado XAML, a fin de responder a los requisitos de los colaboradores de programación de aplicaciones; Microsoft Expression Blend proporciona una experiencia apropiada para los diseñadores, mientras que Visual Studio 2005 está dirigido a los programadores.
- La globalización y localización de las aplicaciones WPF se ha simplificado en gran medida



En el ejemplo siguiente se utiliza XAML para implementar la apariencia de una ventana que contiene un solo botón.

```
C#
<Window
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  Title="Window with Button"
  Width="250" Height="100">

  <!-- Add button to window -->
  <Button Name="button">Click Me!</Button>

</Window>
```

En tiempo de ejecución, WPF convierte los elementos y los atributos definidos en el marcado en instancias de clases de WPF. Por ejemplo, el elemento Window se convierte en una instancia de la clase Windows cuya propiedad Title es el valor del atributo Title.

En la figura 49, se muestra la interfaz de usuario (UI) definida mediante el código XAML del ejemplo anterior.



Figura 49: Interfaz grafica definida mediante XAML.

**3.3.3. Código subyacente.** El comportamiento principal de una aplicación es implementar la funcionalidad que responde a las interacciones con el usuario, lo que incluye controlar los eventos (por ejemplo, hacer clic en un menú, una barra de herramientas o un botón) y llamar, en respuesta, a la lógica empresarial y de acceso a los datos.

En WPF, este comportamiento se suele implementar en código asociado al marcado. Este tipo de código se denomina subyacente. En el ejemplo siguiente se muestran el código subyacente y el marcado actualizado del ejemplo anterior.

```
C#
<Window
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  x:Class="SDKSample.AWindow"
```

```
Title="Window with Button"
Width="250" Height="100">

<!-- Add button to window -->
<Button Name="button" Click="button_Click">Click Me!</Button>

</Window>
```

## C#

```
using System.Windows; // Window, RoutedEventArgs, MessageBox

namespace SDKSample
{
    public partial class AWindow : Window
    {
        public AWindow()
        {
            // InitializeComponent call is required to merge the UI
            // that is defined in markup with this class, including
            // setting properties and registering event handlers
            InitializeComponent();
        }

        void button_Click(object sender, RoutedEventArgs e)
        {
            // Show message box when button is clicked
            MessageBox.Show("Hello, Windows Presentation Foundation!");
        }
    }
}
```

En la siguiente ilustración, figura 50, se muestra el resultado de hacer clic en el botón.



Figura 50: Resultado de la programación del código subyacente de un botón.

### 3.4. WCF WINDOWS COMMUNICATION FOUNDATION

**3.4.1. Introducción.** Windows Communication Foundation (WCF) está diseñado para ofrecer un enfoque manejable a la informática distribuida, interoperabilidad ancha y asistencia directa para la orientación sobre el servicio.

WCF simplifica el desarrollo de aplicaciones conectadas a través de un nuevo modelo de programación orientado a servicios. WCF admite muchos estilos de desarrollo de aplicaciones distribuidas proporcionando una arquitectura superpuesta. En su base, la arquitectura de canal de WCF proporciona primitivos asíncronos de paso de aprobación de mensajes sin tipo. Generados sobre esta base están las funciones de protocolos para un intercambio de datos de transacción seguro y fiable, así como una amplia variedad de opciones de codificación y transporte.

Hay tres cosas que destacan como los aspectos más importantes de WCF:

- Unificación de las tecnologías de comunicación .NET Framework existentes.
- Compatibilidad para interoperabilidad entre proveedores, incluyendo confiabilidad, seguridad y transacciones.
- Orientación explícita al servicio.

Windows Communication Foundation (WCF)WCF es un tiempo de ejecución y un conjunto de API para la creación de sistemas que envíen mensajes entre servicios y clientes. La misma infraestructura y API se utilizan para crear aplicaciones que se comuniquen con otras aplicaciones en el mismo sistema del equipo o en un sistema que resida en otra compañía y a la que se obtenga acceso a través de Internet.

**3.4.2. Mensajería y extremos.** WCF se basa en la noción de comunicación basada en mensajes y cualquier cosa que se pueda modelar como un mensaje (por ejemplo, una solicitud HTTP o un mensaje de MSMQ), se puede representar de manera uniforme en el modelo de programación. Esto habilita una API unificada en todos los mecanismos de transporte diferentes.

El modelo distingue entre **clientes**, que son aplicaciones que inician la comunicación y **servicios**, que son aplicaciones que esperan a que los clientes se comuniquen con ellos y respondan a esa comunicación. Una única aplicación puede actuar como cliente y como servicio.

Los mensajes se envían entre extremos. Los extremos son los lugares donde los mensajes se envían o reciben (o ambos), y definen toda la información requerida para el intercambio de mensajes. Un servicio expone uno o más extremos de la

aplicación (y a cero o más extremos de la infraestructura), y el cliente genera un extremo que es compatible con uno de los extremos del servicio.

Un **extremo** describe de una manera basada en estándar dónde se deberían enviar los mensajes, cómo se deberían enviar y qué aspecto deberían tener los mensajes. Un servicio puede exponer esta información como metadatos que los clientes pueden procesar para generar clientes WCF adecuados y **pilas** de comunicación.

**3.4.3. Protocolos de comunicaciones.** Un elemento requerido de la pila de la comunicación es el **protocolo de transporte**. Los mensajes se pueden enviar a través de intranets e Internet utilizando transportes comunes, como HTTP y TCP. Otros transportes incluidos admiten la comunicación con aplicaciones Message Queue Server de Microsoft (MSMQ) y nodos en una malla de redes del mismo nivel. Se pueden agregar más mecanismos de transporte utilizando los puntos de la extensión integrados de WCF.

Otro elemento necesario en la pila de comunicación es la codificación que especifica cómo se da formato a cualquier mensaje determinado. WCF proporciona las siguientes codificaciones:

- Codificación de texto, una codificación interoperable.
- Codificación Mecanismo de optimización de transmisión de mensajes (MTOM), que es una manera interoperable de enviar eficazmente datos binarios no estructurados a y desde un servicio.
- Codificación binaria para una transferencia eficaz.

Se pueden agregar más mecanismos de codificación (por ejemplo, una codificación de compresión) utilizando los puntos de extensión integrados de WCF.

### **3.5. MICROSOFT VISUAL ESTUDIO.**

Microsoft Visual Estudio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.

Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión net 2002). Así se pueden crear aplicaciones que se comuniquen entre estaciones de trabajo, páginas web y dispositivos móviles.

**3.5.1 Visual Studio 2008.** Visual Studio 2008 fue publicado (RTM) el 17 de Noviembre de 2007 en inglés, mientras que la versión en castellano no fue publicada si no hasta el 2 de Febrero de 2008.

El nuevo framework (.Net 3.5) está diseñado para aprovechar las ventajas que ofrece el nuevo sistema operativo "Windows Vista" a través de sus subsistemas "Windows Communication Foundation" (WCF) y "Windows Presentation Foundation" (WPF). El primero tiene como objetivo la construcción de aplicaciones orientadas a servicios mientras que el último apunta a la creación de interfaces de usuario más dinámicas que las conocidas hasta el momento.

A las mejoras de desempeño, escalabilidad y seguridad con respecto a la versión anterior, se agregan entre otras, las siguientes novedades:

- La mejora en las capacidades de Pruebas Unitarias permiten ejecutarlas más rápido independientemente de si lo hacen en el entorno IDE o desde la línea de comandos. Se incluye además un nuevo soporte para diagnosticar y optimizar el sistema a través de las herramientas de pruebas de Visual Studio. Con ellas se podrán ejecutar perfiles durante las pruebas para que ejecuten cargas, prueben procedimientos contra un sistema y registren su comportamiento; y utilizar herramientas integradas para depurar y optimizar.
- Con Visual Studio Tools for Office (VSTO), integrado con Visual Studio 2008 es posible desarrollar rápidamente aplicaciones de alta calidad basadas en la interfaz de usuario (UI) de Office que personalicen la experiencia del usuario y mejoren su productividad en el uso de Word, Excel, PowerPoint, Outlook, Visio, InfoPath y Project. Una completa compatibilidad para implementación con ClickOnce garantiza el entorno ideal para una fácil instalación y mantenimiento de las soluciones Office.
- Visual Studio 2008, permite incorporar características del nuevo Windows Presentation Foundation sin dificultad tanto en los formularios de Windows existentes como en los nuevos. Ahora es posible actualizar el estilo visual de las aplicaciones al de Windows Vista debido a las mejoras en Microsoft Foundation Class Library (MFC) y Visual C++. Visual Studio 2008 permite mejorar la interoperabilidad entre código nativo y código manejado por .NET. Esta integración más profunda simplificará el trabajo de diseño y codificación.
- LINQ (Language Integrated Query) es un nuevo conjunto de herramientas diseñado para reducir la complejidad del acceso a Base de Datos, a través de extensiones para C++ y Visual Basic así como para Microsoft .NET Framework. Permite filtrar, enumerar, y crear proyecciones de muchos tipos y colecciones de datos utilizando todos la misma sintaxis, prescindiendo del uso de lenguajes especializados como SQL o XPath.

- Visual Studio 2008 ahora permite la creación de soluciones multiplataforma adaptadas para funcionar con las diferentes versiones de .Net Framework: 2.0. (Incluido con Visual Studio 2005), 3.0 (incluido en Windows Vista) y 3.5 (incluido con Visual Studio 2008).
- .NET 3.5 incluye biblioteca ASP.NET AJAX para desarrollar aplicaciones web más eficientes, interactivas y altamente personalizadas que funcionen para todos los navegadores más populares y utilicen las últimas tecnologías y herramientas Web, incluyendo Silverlight y Popfly.

### 3.6. EXPRESSION STUDIO

**Expression Studio** es una suite de herramientas de diseño profesional y contenido multimedia desarrollado por Microsoft, que proporcionan flexibilidad y libertad para elaborar contenidos multimedia de manera sencilla.

**3.6.1. Expression Studio 2.** Expression Studio 2, es la versión más reciente de esta suite la cual permite llevar a la creatividad a un nuevo nivel, en esta versión se incluye herramientas de diseño y tecnologías innovadoras adicionales al Expresión Studio inicial, dándole flexibilidad y libertad para llevar su visión a la realidad, esta nueva suite incluye 5 aplicaciones:

- Expression Web 2
- Expression Blend 2
- Expression Design 2
- Expression Media 2
- Expression Encoder 2.

**3.6.1.1 Expression Blend 2.** Expression Blend 2, es su herramienta para aprovechar la potencia de XAML, .NET Framework 3,5 y Silverlight para obligar a entregar la mejor experiencia al usuario de ordenadores personales conectados y la red.

Expression Blend 2, en esta versión, hay mejoras significativas en la interoperabilidad con XAML, y Silverlight.

- Mejoras con Silverlight: Expression Blend 2 soporta proyectos Silverlight 1.0, que son creados como proyectos de sitio web en lugar de proyectos de aplicación, esto ayuda a mejora la interoperabilidad con Visual Studio 2008, pudiendo modificar un proyecto Silverlight 1.0 en cualquier momento, Expression Blend 2 incluye un editor de texto de JavaScript para editar los códigos Silverlight 1,0 detrás de los archivos.

- Mejoras con Visual Studio 2008: Abrir y trabajar con proyectos de Visual Studio 2008.
- Cambios en los archivos sincronizados: Al modificar un proyecto Silverlight en Expression Blend 2 y Visual Studio al mismo tiempo, cualquier cambio que se haga en Visual Studio se aplicará inmediatamente en Expression Blend 2.
- Storyboard Picker, Sustituye a la antigua caja de Storyboard , picker se compone de una etiqueta que indique el nombre del Storyboard seleccionado (un guion, si se selecciona), pudiendo acceder a este a través del menú de acceso directo (disponible cuando usted haga clic derecho en la etiqueta), un botón de atajo (y la consiguiente menú contextual), un botón cerrar, para cerrar todos los storyboards además de la salida modo de grabación. Tanto el menú y el menú permiten crear guiones, y si un guion ya está seleccionado, se puede duplicar, revertir, o eliminar.
- Clip camino de edición y animación, Expression Blend 2 ahora permite modificar la saturación en los arboard después de crearlos, pudiendo animar cada uno de los vértices de saturación.

## 4. ANALISIS, DISEÑO E IMPLEMENTACION DE FACEMATH

### 4.1. ANALISIS DE FACEMATH

Para el análisis de FaceMath, se construyen los siguientes artefactos:

- Tabla de funciones o requerimientos.
- Diagramas de casos de uso.

**4.1.1. Funciones.** Las funciones o requerimientos que se tuvieron en cuenta para el desarrollo de FaceMath están plasmados en la tabla 3.

Las funcionalidades del Servidor son todas las mostradas a excepción de R2.2 y R7 que son del cliente.

Tabla 3: Tabla de funciones de FaceMath.

Ref. #	Función	Categoría	Atributo	Detalles y Restricciones	Categoría
R1	Arquitectura Cliente-Servidor	Evidente	Arquitectura		Obligatorio
R2	Iniciar Prototipo	Evidente	Interfaz	Amigable al usuario	Obligatorio
R2.1	Iniciar Interfaz grafica del Servidor	Evidente	Interfaz		Obligatorio
R2.1.1	Mostrar interfaz grafica del servidor	Evidente	Interfaz	Animación de Ventana	Obligatorio
R2.2	Iniciar interfaz grafica del Cliente	Evidente	Interfaz		Deseable
R3	Mostrar mensajes de error cuando algo no esté bien	Evidente	Interfaz	Cuadros de Dialogo	Deseable
R4	Permitir Configuración del prototipo	Evidente	Interfaz	Amigable	Obligatorio
R4.1	Permitir seleccionar directorio principal	Evidente	Interfaz		Obligatorio
R4.2	Permitir ajustar puerto de servicio	Evidente	Interfaz		Deseable
R4.3	Permitir ajustar atributos de imagen	Evidente	Interfaz		Deseable



R4.4	Permitir ajustar umbral de reconocimiento	Evidente	Interfaz		Obligatorio
R4.5	Permitir guardar cambios	Evidente	Interfaz		Opcional
R4.5.1	Cargar archivo y guardar cambios	Oculto	Información		Obligatorio
R5	Permitir Administración de Usuarios	Evidente	Interfaz	Amigable	Obligatorio
R5.1	Permitir agregar usuario	Evidente	Interfaz		Opcional
R5.2	Permitir borrar usuario	Evidente	Interfaz		Opcional
R5.3	Permitir actualizar usuario	Evidente	Interfaz		Opcional
R5.4	Permitir ver detalles del usuario	Evidente	Interfaz		Obligatorio
R5.4.1	Cargar archivo y guardar cambios	Oculto	Información		Obligatorio
R6	Permitir Administración del Servicio	Evidente	Interfaz	Amigable	Obligatorio
R6.1	Permitir Iniciar el servicio	Evidente	Interfaz		Obligatorio
R6.1.1	Iniciar servicio WCF en la maquina	Oculto	Comunicación		Obligatorio
R6.2	Permitir detener el servicio	Evidente	Interfaz		Opcional
R6.2.1	Detener servicio WCF en la maquina	Oculto	Comunicación		Opcional
R6.3	Permitir ver log del servicio	Evidente	Interfaz		Obligatorio
R6.3.1	Cargar archivo	Oculto	Información		Obligatorio
R6.4	Permitir eliminar log del servicio	Evidente	Interfaz		Opcional
R6.3.1	Cargar archivo y eliminar log	Oculto	Información		Opcional
R7	Permitir reconocimiento	Evidente	Interfaz	Amigable	Opcional
R7.1	Iniciar conexión con el servidor	Oculto	Comunicación		Obligatorio
R7.2	Permitir reconocimiento desde archivo	Evidente	Interfaz		Opcional
R7.2.1	cargar archivo imagen y envirlarla al servidor	Oculto	Comunicación		Opcional

R7.3	Permitir reconocimiento desde WebCam	Evidente	Interfaz	Amigable	Opcional
R7.3.1	Capturar imagen y enviarla al servidor	Oculto	Comunicación		Opcional
R7.4	Mostrar resultado de reconocimiento	Evidente	Interfaz	Amigable	Obligatorio

#### 4.1.2. Diagramas de Casos de Uso

Figura 51: Diagrama de Caso de Uso general.

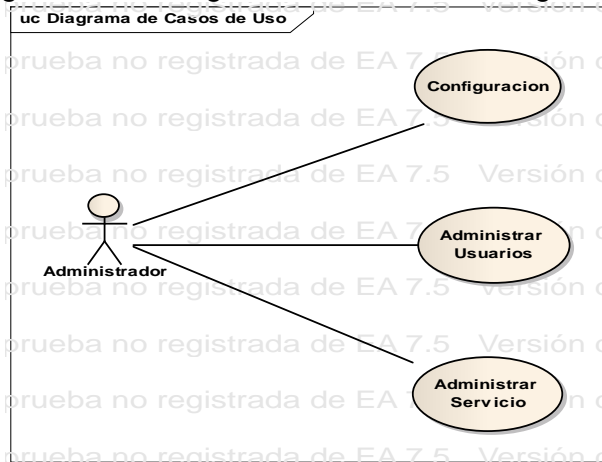


Figura 52: Diagrama de Caso de Uso, Modulo Configuración.

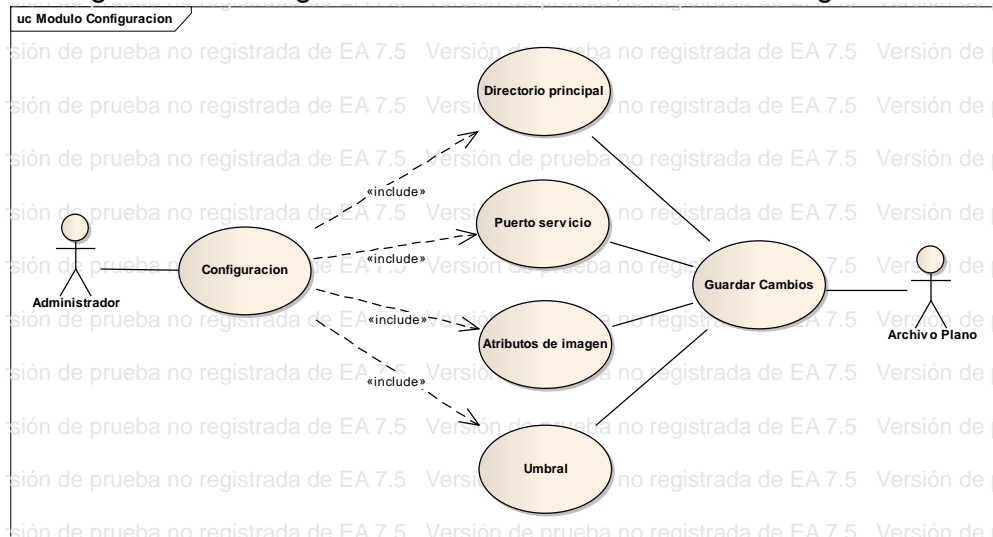


Figura 53: Diagrama de Caso de Uso, Modulo Administración Usuarios.

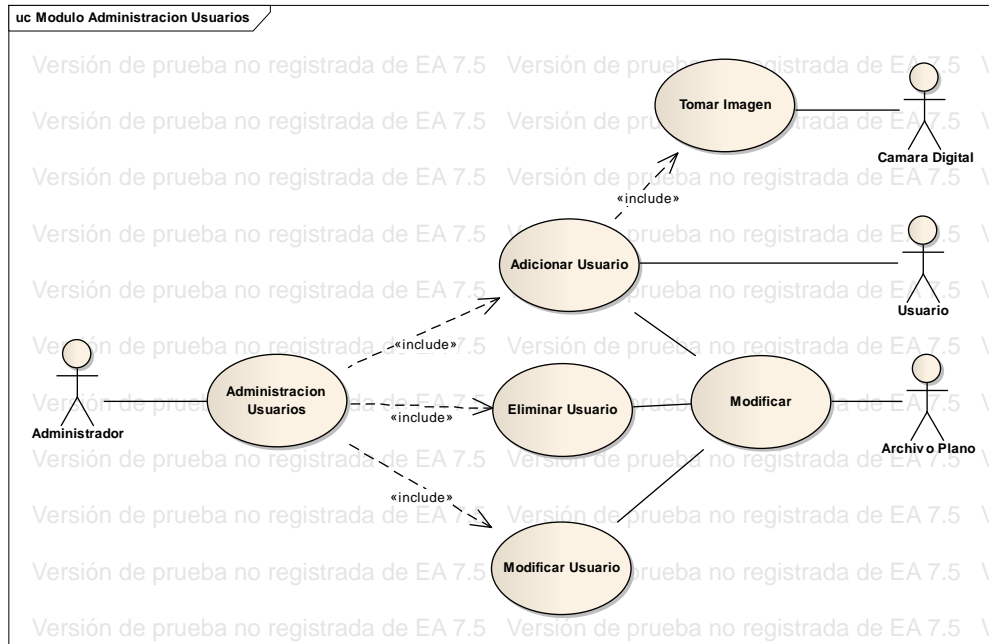
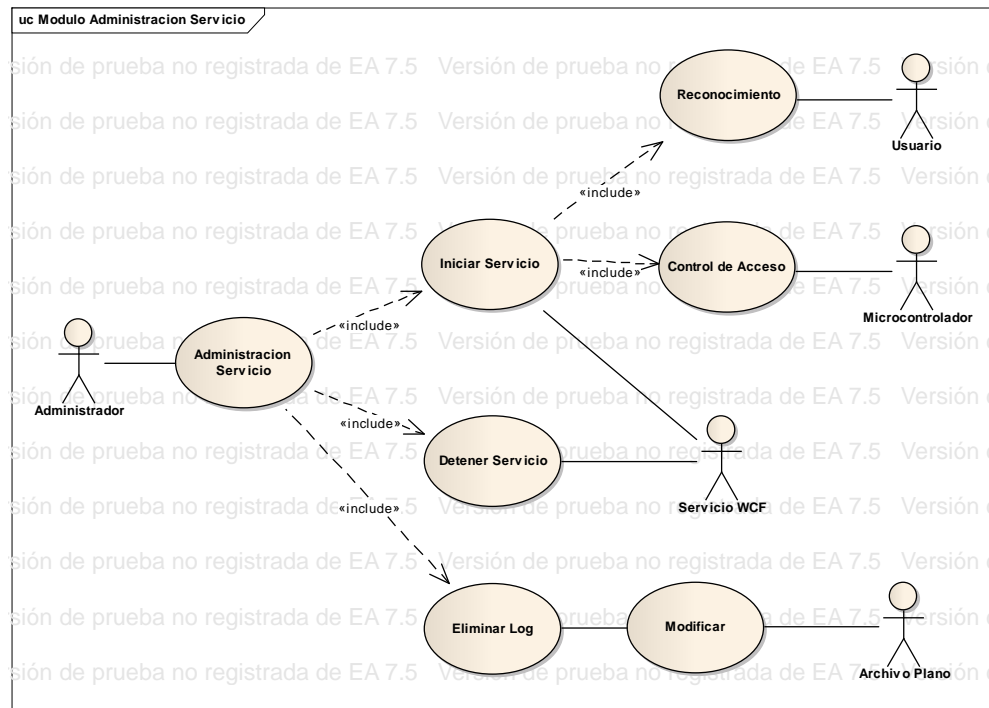


Figura 54: Diagrama de Caso de Uso, Modulo Administración Servicio.



## 4.2 DISEÑO

Para el diseño de FaceMath se hace uso de los siguientes artefactos:

- Diagrama de Paquetes.
- Diagrama de Clases.
- Diagrama de Secuencia.

### 4.2.1. Diagramas de Paquetes.

Figura 55: Diagrama de paquetes principal.

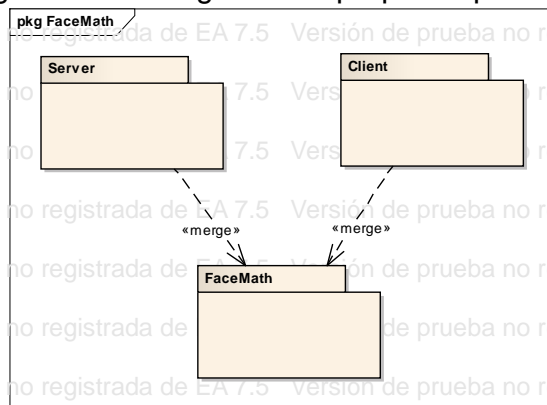
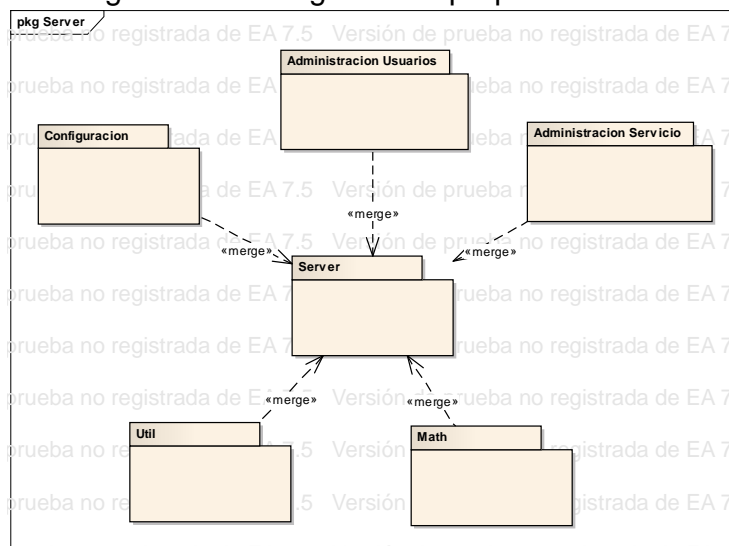


Figura 56: Diagrama de paquete Server.



## 4.2.2. Diagrama de Clases

Figura 57: Paquete Math.

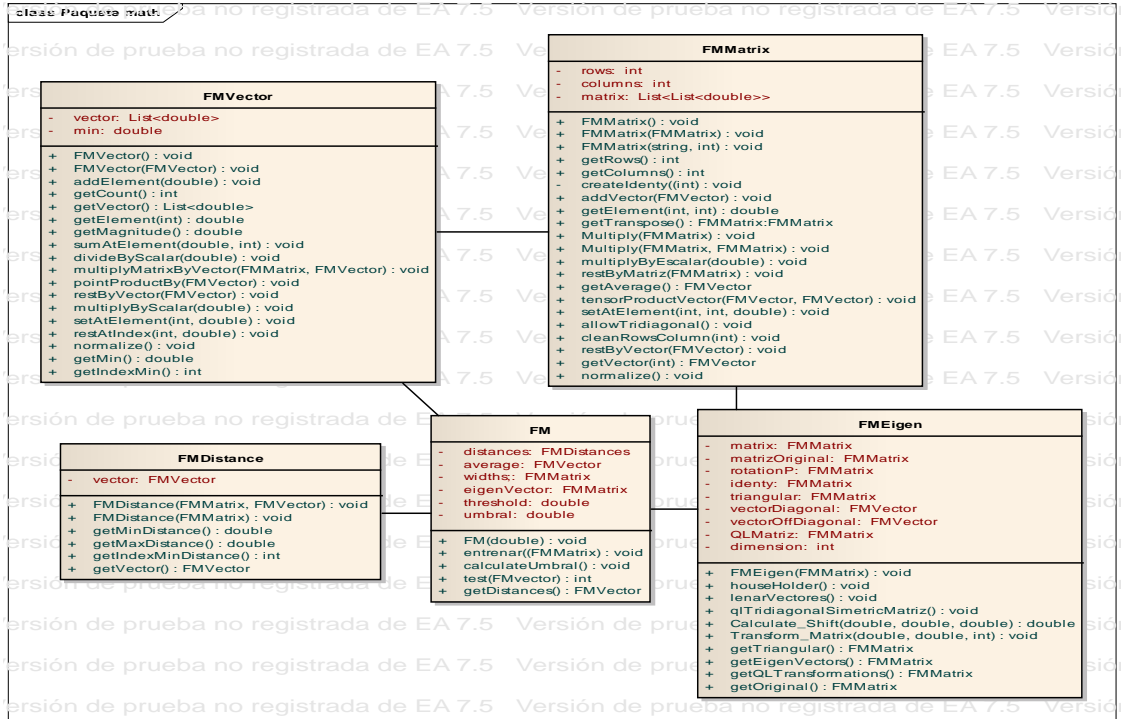


Figura 58: Paquete Administración Servicio.

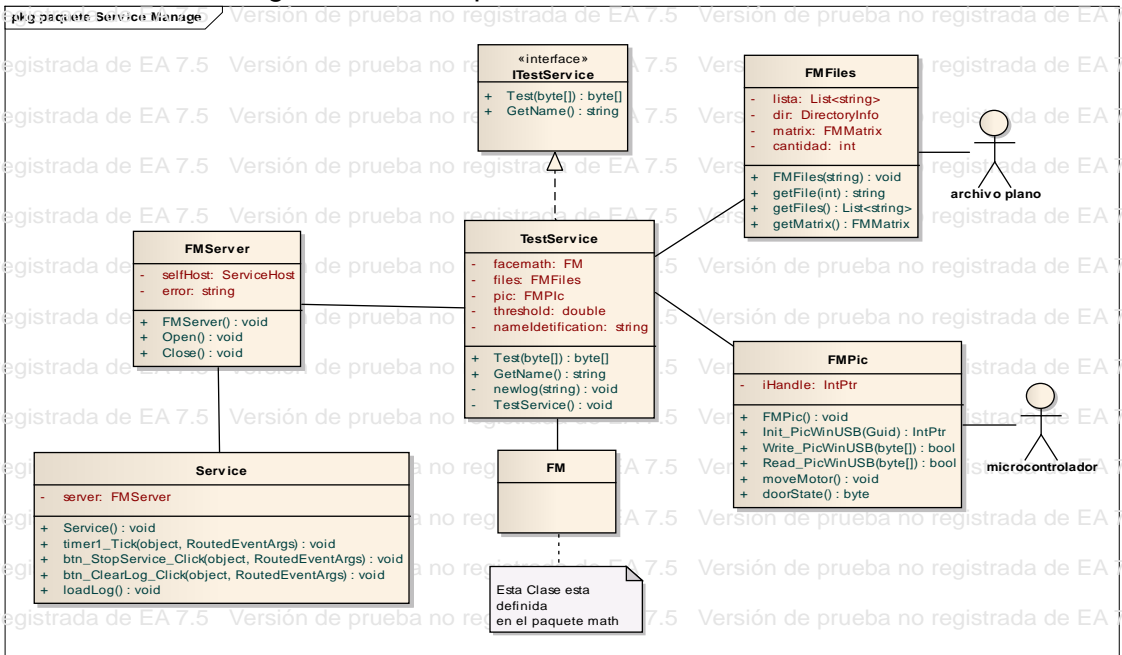


Figura 59: Paquete Administración Usuarios.

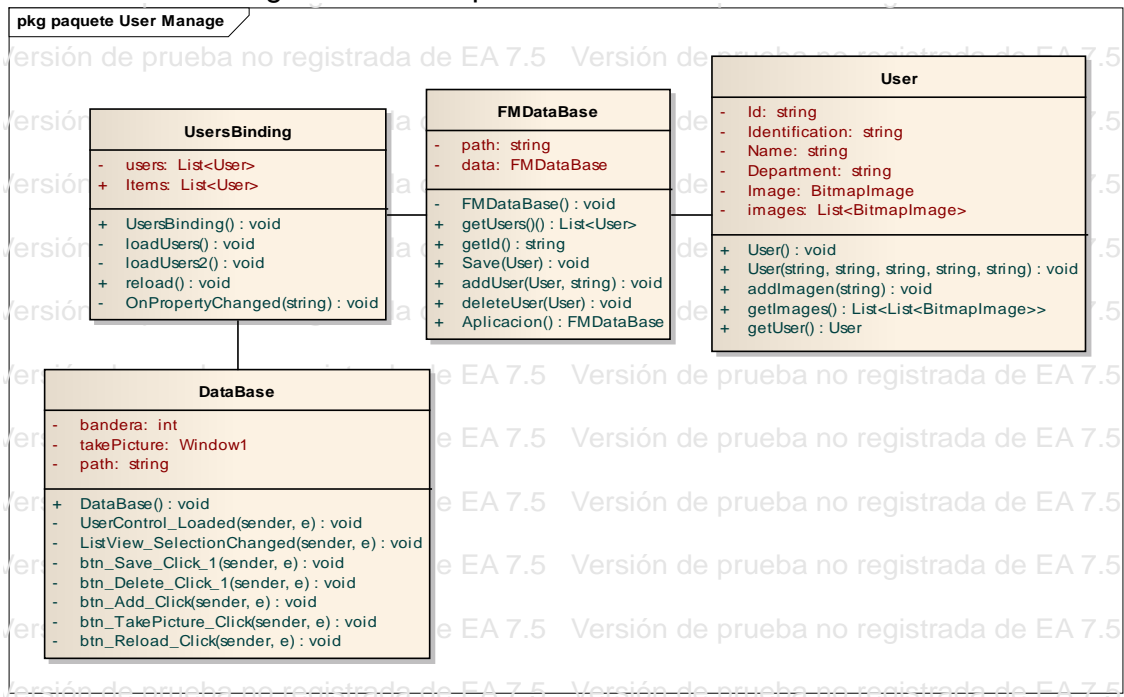


Figura 60: Paquete Configuración.

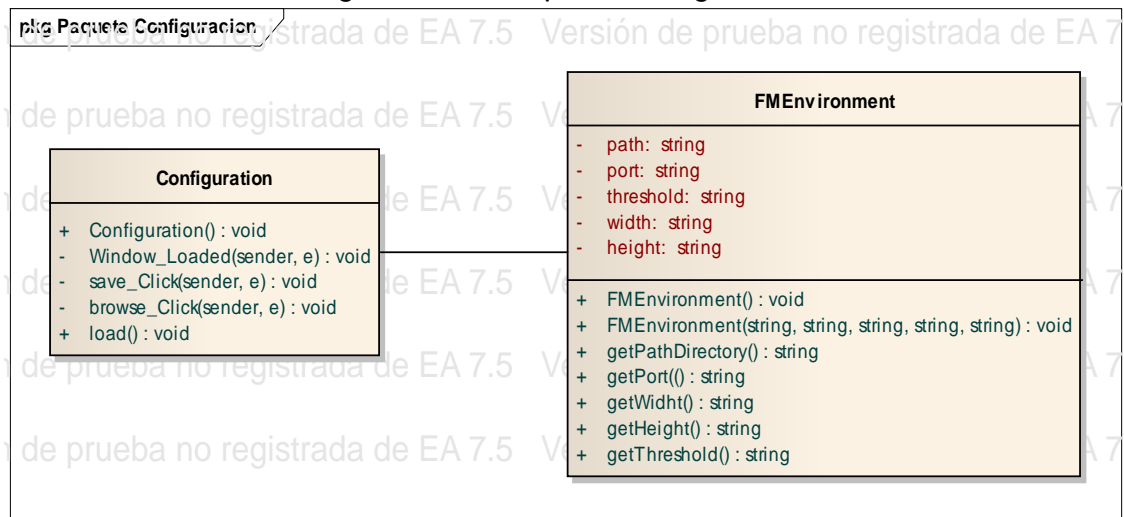
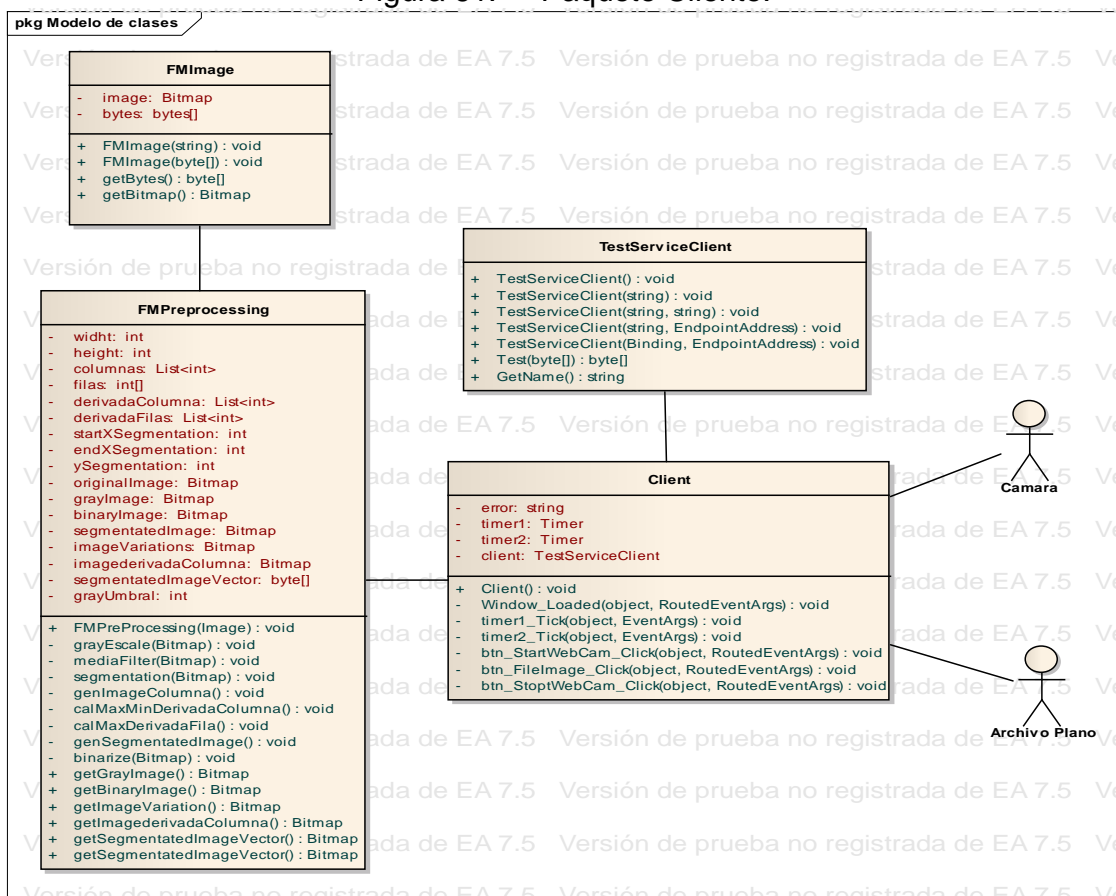


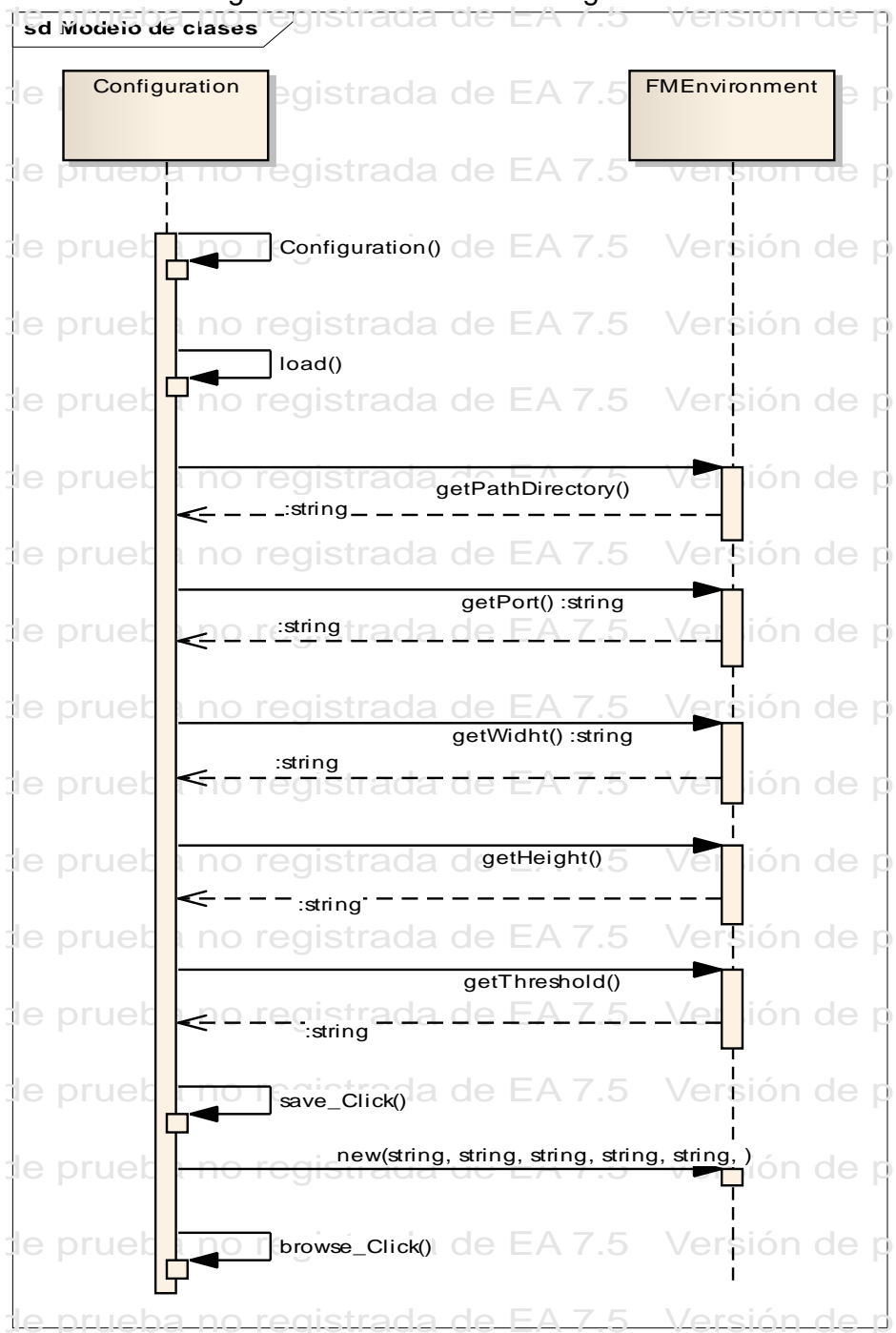
Figura 61: Paquete Cliente.



#### 4.2.3. Diagrama de Secuencia

## Clase Configuration.

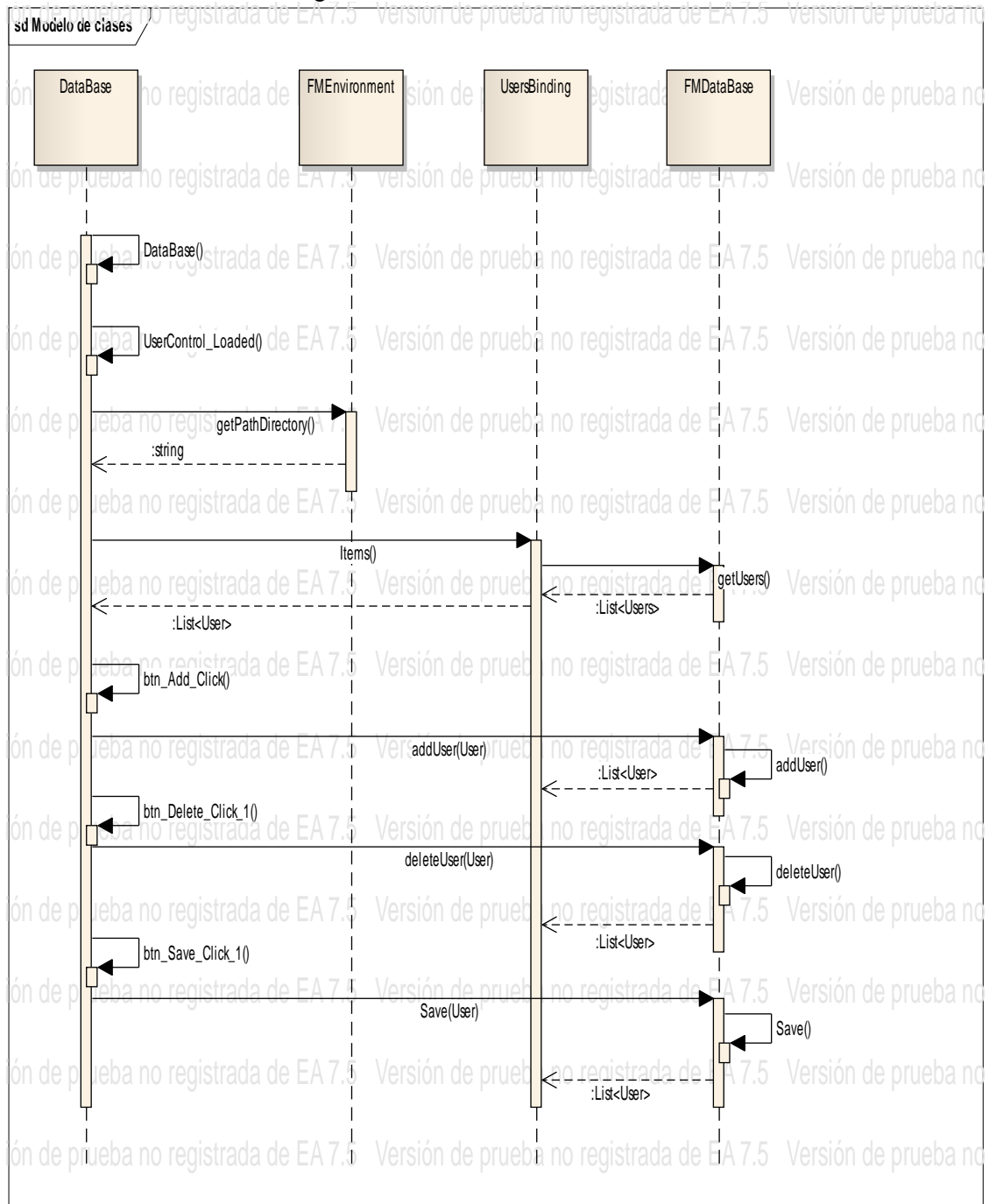
Figura 62: run Class Configuration.





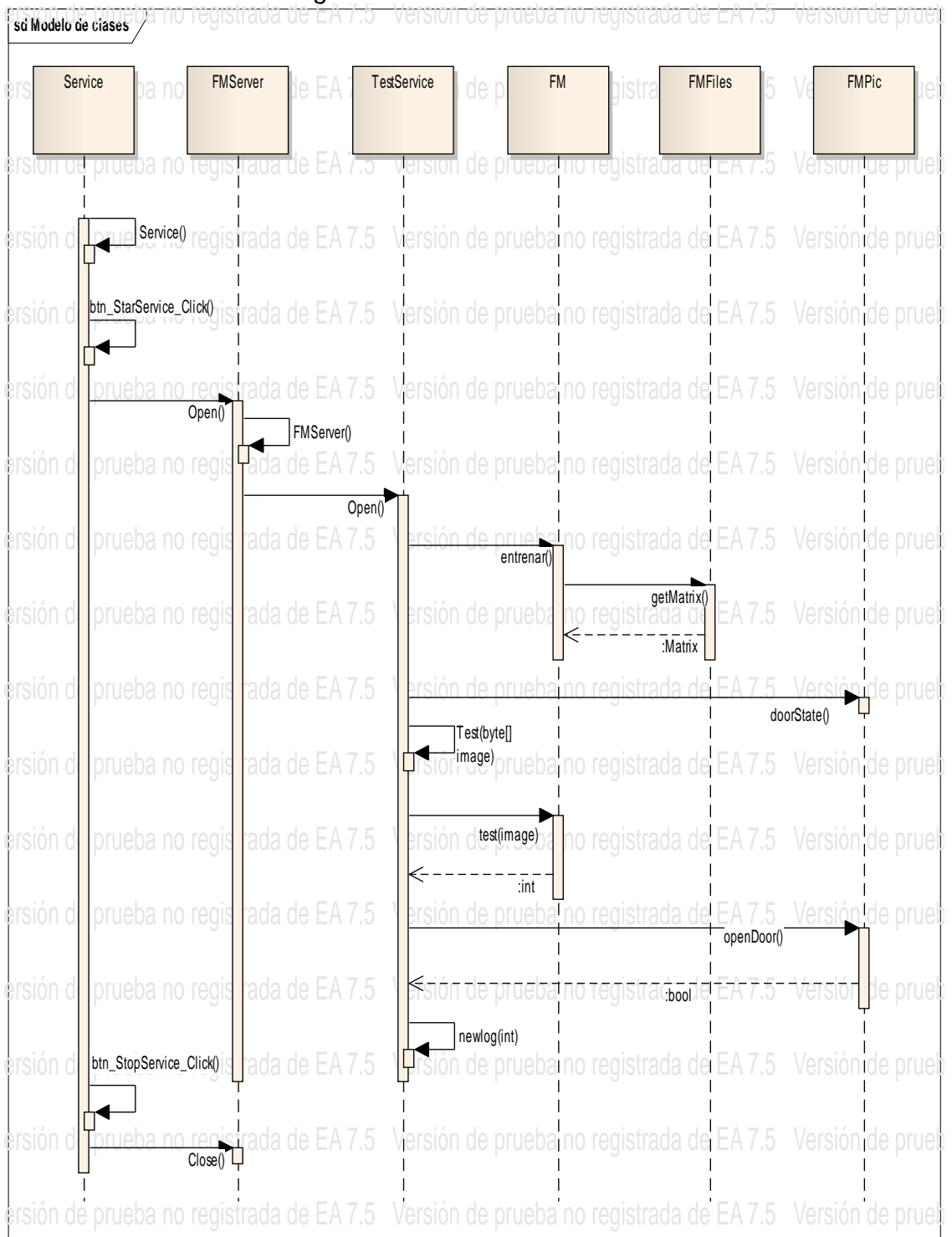
## Clase DataBase.

Figura 63: run Clase DataBase.



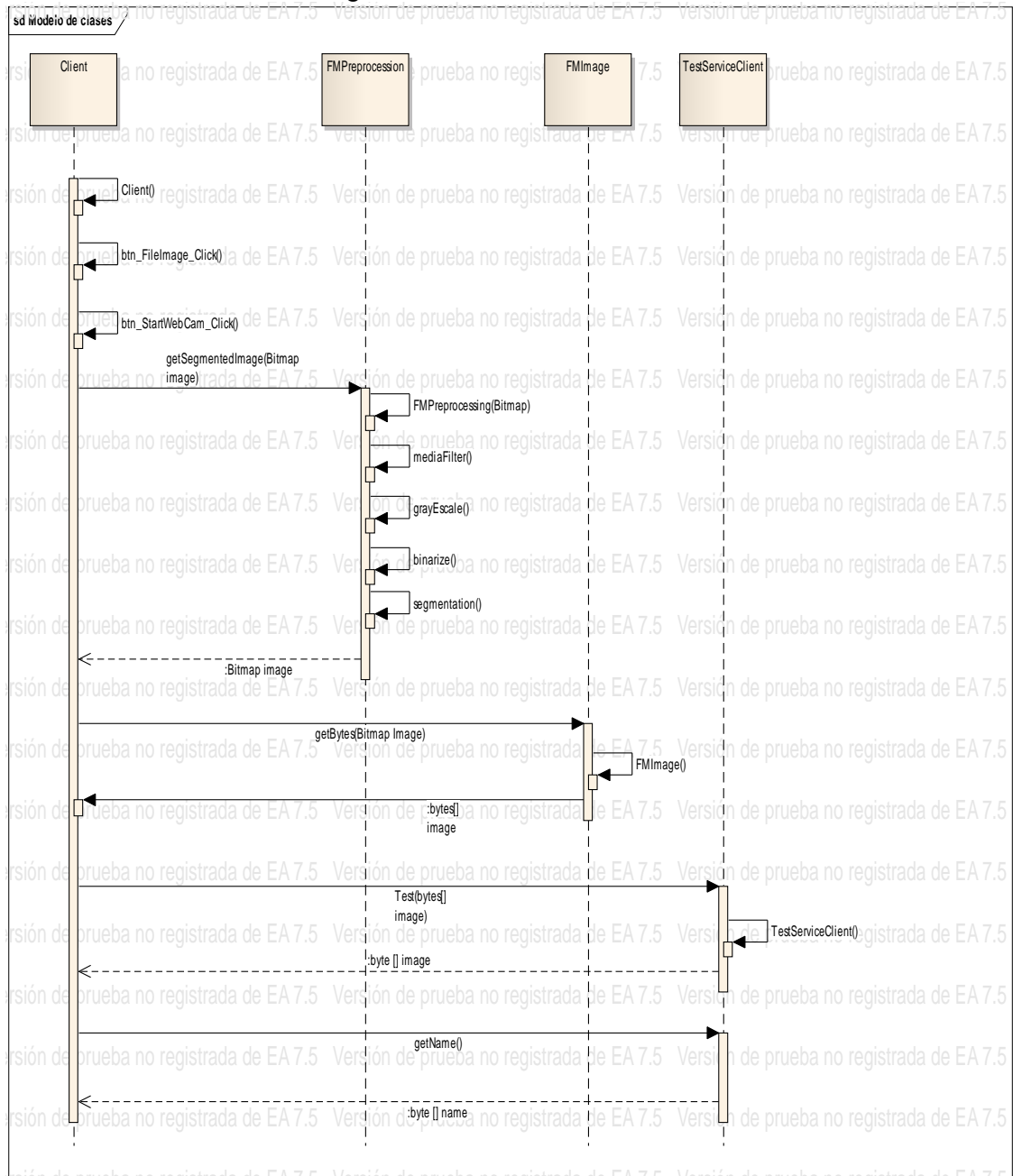
## Clase Service.

Figura 64: run Clase Service.



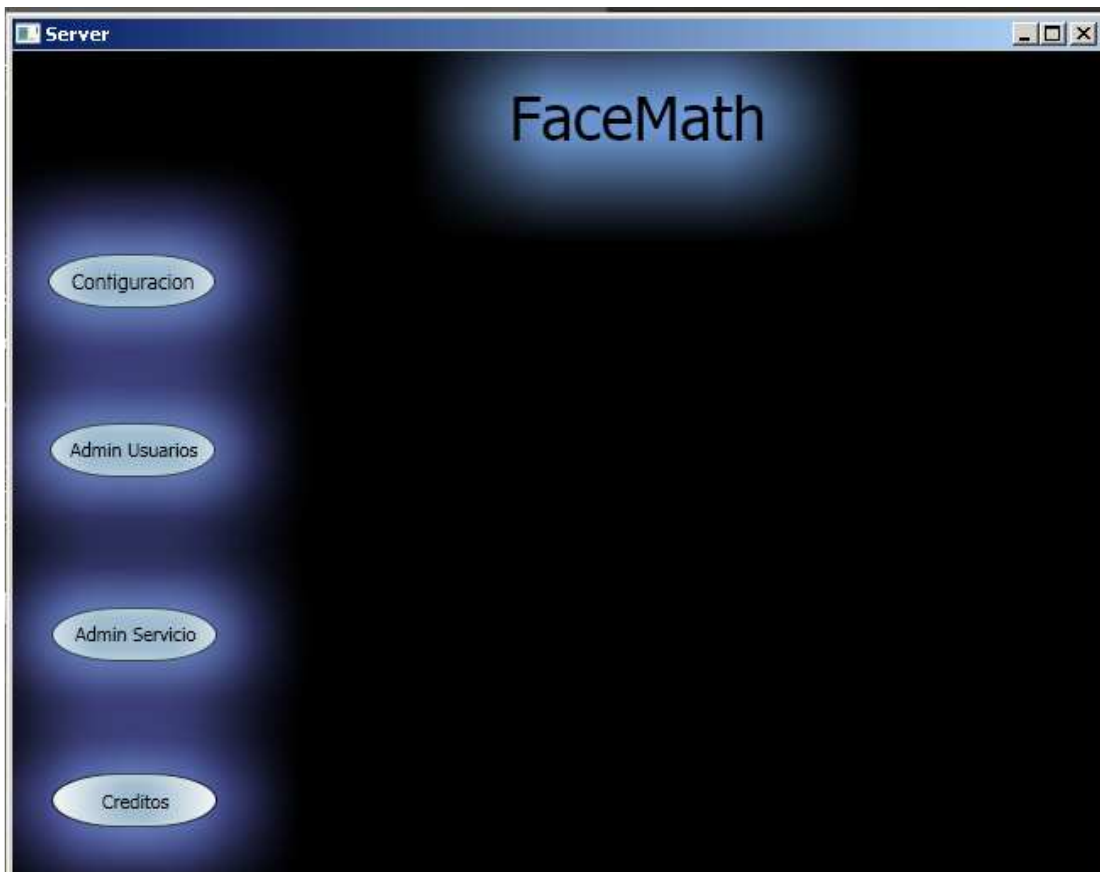
## Clase Client.

Figura 65: Run Class Client



### 4.3. IMPLEMENTACION

Figura 66: Interfaz gráfica del Servidor.

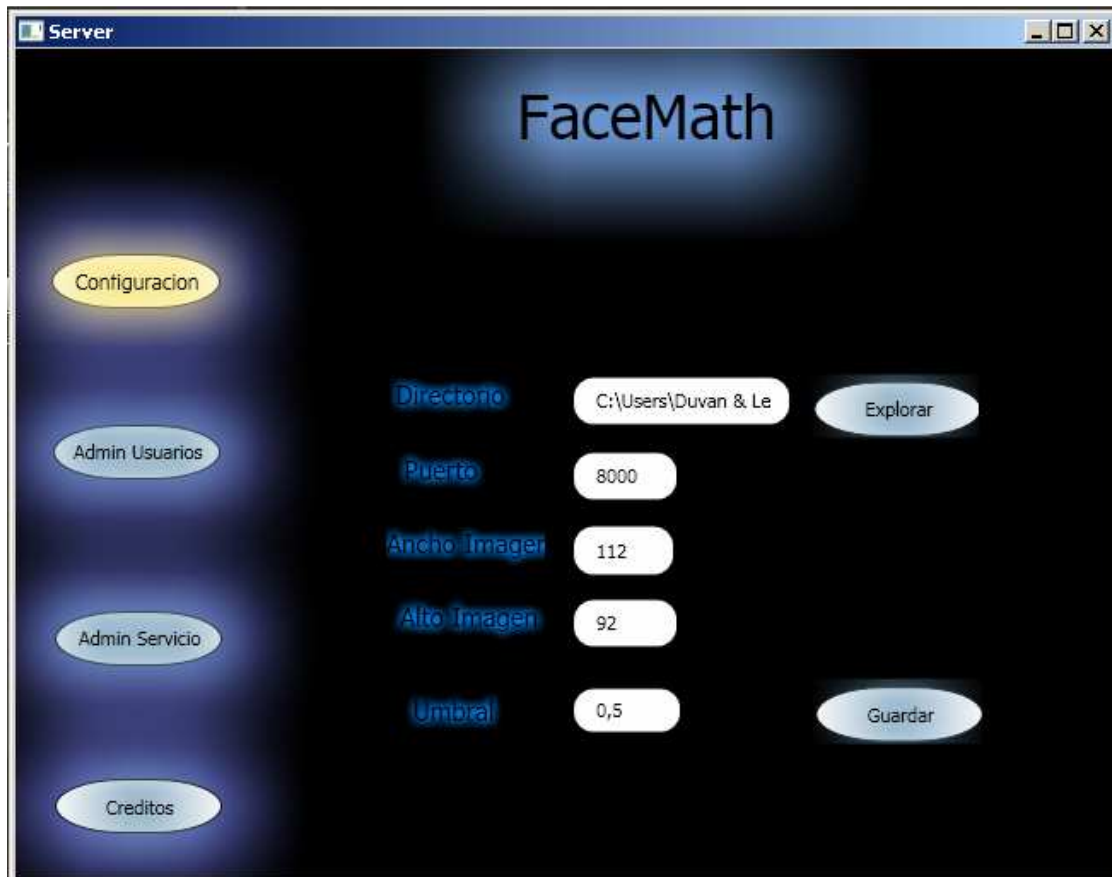


Esta es la pantalla principal del Servidor.

Se puede apreciar que es una ventana muy llamativa, donde la forma de los botones es diferente a la tradicional. Todo esto gracias a WPF.

En el servidor se encuentran los módulos de configuración, Administración de Usuarios y Administración del Servicio.

Figura 67: Interfaz gráfica módulo Configuración.

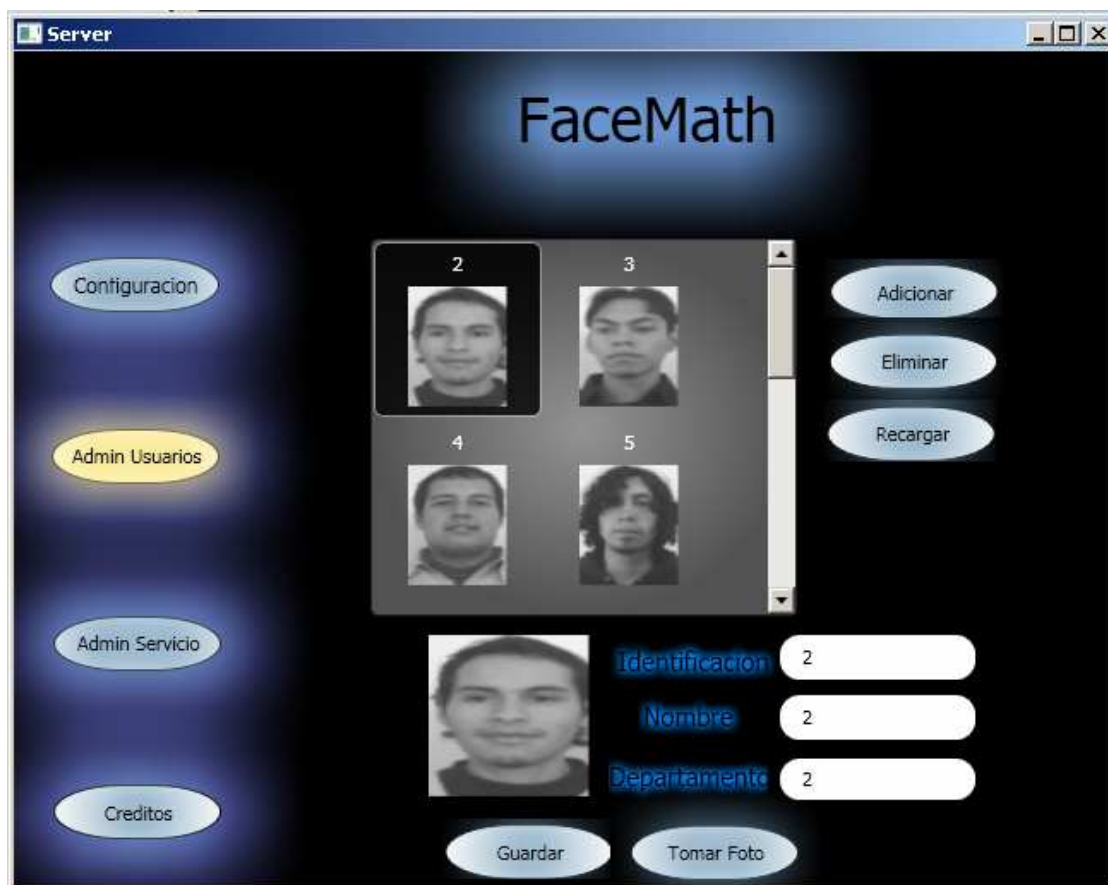


En el módulo de configuración, los parámetros más importantes son “Directorio” y “umbral”.

En Directorio se indica en que carpeta del sistema se van almacenar las imágenes e información de los usuarios. Hay que tener en cuenta que es una ruta absoluta, por lo que al ejecutar el programa por primera vez hay que indicar este parámetro.

El parámetro umbral es el error permitido por el sistema en el proceso de reconocimiento, cuyo valor varía entre 0,0 (no reconoce a nadie) y 1,0 (cualquier imagen es reconocida).

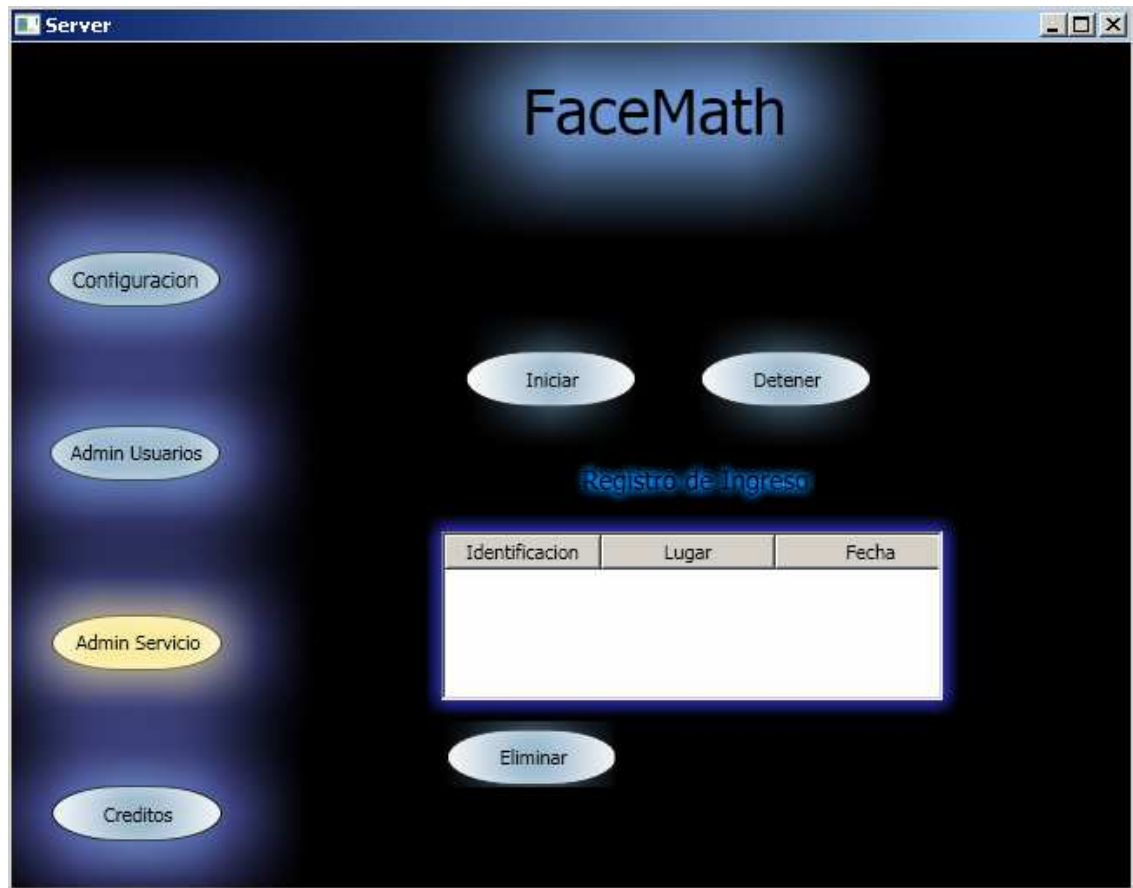
Figura 68: Interfaz gráfica Módulo Administración de Usuarios.



Como se puede observar, en este módulo se administra la información de los usuarios, se puede adicionar, eliminar o modificar un usuario; es aquí también donde se toma la fotografía.

Se ha personalizado el objeto ListView de WPF para que sea contenedor de imágenes y títulos. Se utiliza Binding para que la información de las cajas de texto muestre la información del usuario seleccionado.

Figura 69: Interfaz gráfica Módulo Administración del Servicio.



Este es el módulo de Administración del Servicio.

En este módulo se puede habilitar o deshabilitar el control de acceso a través de los botones "iniciar" y "detener".

Cuando un usuario es reconocido, el sistema registra que usuario es, a qué lugar ingresa y en qué momento.

Estos registros se pueden observar en el log del servicio.

Figura 70: Interfaz gráfica Módulo Cliente.



Y finalmente el Cliente, en donde el usuario debe pasar por el lector para ser reconocido.

Se puede observar que existen dos opciones de reconocimiento. La primera a través de la cámara. La segunda a través de un archivo, muy útil cuando se realizaron las pruebas.



## 5. PRUEBAS Y RESULTADOS

Para las siguientes pruebas se utilizó un computador portátil Dell modelo Inspiron I1420, procesador Intel Core 2 Duo 1.5 GHz, memoria RAM 1 GB, Sistema Operativo Windows Vista Home Basic, service pack 1.

Los resultados que se puede obtener del reconocimiento, son cuatro:

- Verdadero-Positivo: Cuando se presenta una imagen del rostro de una persona que está en la base de datos y el sistema identifica correctamente quien es. Ejemplo:



Figura 71: Resultado del reconocimiento como Verdadero-Positivo.

- Falso-Positivo: Cuando se presenta una imagen del rostro de una persona que no está en la base de datos y el sistema lo reconoce erróneamente. Ejemplo:



Figura 72: Resultado del reconocimiento como Falso-Positivo.

- Verdadero-Negativo: Cuando se presenta una imagen del rostro de una persona que no está en la base de datos y el sistema lo rechaza. Ejemplo:



Figura 73: Resultado del reconocimiento como Verdadero-Negativo.

- Falso-Negativo: Cuando se presenta una imagen del rostro de una persona que está en la base de datos y el sistema lo rechaza. Ejemplo:



Figura 74: Resultado del reconocimiento como Falso-Negativo.

Para las pruebas 5.1 y 5.2 se utilizó la base datos de AT & T, ver anexo 1.

La prueba 5.3 se utilizó con una base de datos propia, generada por el sistema, ver anexo 2.

## 5.1. MANTENER EL UMBRAL, VARIAR EL NÚMERO DE USUARIOS

**5.1.1. Prueba 1.1: umbral 0.2, 10 usuarios.** Se tomó la primera imagen de 10 sujetos de la base de datos de AT & T y el resto fue utilizada como prueba, el umbral fue de 0,2, los resultados fueron los siguientes:

Sujeto	Imágenes Reconocidas	Imágenes No Reconocidas
1	ninguna	todas
2	3,5	2,4,6,7,8,9,10
3	2,3	4,5,6,7,8,9,10
4	3,8	2,4,5,6,7,9,10
5	4,7	2,3,5,6,8,9,10
6	2,3,4,6,7,8,10	5,9
7	3,4	2, 5,6,7,8,9,10
10	ninguna	todas
11	3,10	2,4,5,6,7,8,9
12	6,8	2,3,4,5,6,7,9

Tabla 4: Diseño de la prueba 1.1.

**5.1.2. Prueba 1.2: umbral 0.2, 15 usuarios.** Se tomó la primera imagen de 15 sujetos de la base de datos de AT & T y el resto fue utilizada como prueba, el umbral fue de 0,2, los resultados fueron los siguientes:

Sujeto	Imágenes Reconocidas	Imágenes No Reconocidas
1	ninguna	Todas
2	3,8	2,3,4,6,7,9,10
3	2,3	4,5,6,7,8,9,10
4	3,8	2,4,5,6,7,9,10
5	4,7	2,3,5,6,8,9,10
6	2,3,4,6,8,9,10	5,7
7	3,4	2, 5,6,7,8,9,10
10	ninguna	Todas
11	3,10	2,4,5,6,7,8,9
12	8	2,3,4,5,6,7,8,9
13	2,10	3,4,5,6,7,8,9
14	2,3,4,5	6,7,8,9,10
15	3,4,6	2,5,7,8,9,10
16	ninguna	Todos
17	2,5	3,4,6,7,8,9,10

Tabla 5: Diseño de la prueba 1.2.

**5.1.3. Prueba 1.3: umbral 0.2, 20 usuarios.** Se tomó la primera imagen de 20 sujetos de la base de datos de AT & T y el resto fue utilizada como prueba, el umbral fue de 0,2, los resultados fueron los siguientes:

Sujeto	Imágenes Reconocidas	Imágenes No Reconocidas
1	Ninguna	todas
2	Ninguna	todas
3	2,3	4,5,6,7,8,9,10
4	3,8	2,4,5,6,7,9,10
5	4,7	2,3,5,6,8,9,10
6	2,3,4	5,7,6,8,9,10
7	3,4	2, 5,6,7,8,9,10
10	Ninguna	todas
11	3,10	2,4,5,6,7,8,9
12	8	2,3,4,5,6,7,8,9
13	2	3,4,5,6,7,8,9,10
14	2,3,4,5	6,7,8,9,10
15	3,4,6	2,5,7,8,9,10
16	Ninguna	todos
17	5	2,3,4,6,7,8,9,10
18	2	3,4,5,6,7,8,9,10
19	2,3,10	4,5,6,7,8,9
20	2,10	3,4,5,6,7,8,9
21	2,3,6,7,8,9	4,5,10
22	2,3,5,7,8,9	4,6,10

Tabla 6: Diseño de la prueba 1.3

#### 5.1.4. Análisis de resultados prueba 1.

Prueba	Verdadero Positivo	Falso Negativo	Verdadero Negativo	Falso Positivo	Efectividad
1.1	23.33%	76.67%	100%	0%	61.66%
1.2	22.96%	77.04%	100%	0%	61.48%
1.3	22.77%	77.23%	100%	0%	61.38%

Tabla 7: resultados de la primera 1.

Se observa que el porcentaje de verdadero positivo va disminuyendo. En todas las pruebas el porcentaje de falso positivo es 0, es decir, que el sistema no permitió que ninguna persona que no estuviera en la base datos ingresara al sistema, por lo que cumple con el objetivo de un sistema de control de acceso

Se plantea la hipótesis que las personas no reconocidas se presentan cuando el sujeto tiene un fondo casi completamente negro, como es el caso del sujeto 1 y 10. No se reconoce tampoco sujetos con cuyas imágenes presenten condiciones de luz diferentes a la del entrenamiento como es el caso del sujeto 16. Cuando un

sujeto utiliza gafas, como en el caso del sujeto 17, las imágenes no reconocidas se presentan cuando el sujeto no utiliza gafas.

## 5.2. VARIAR EL UMBRAL, MANTENER EL NÚMERO DE USUARIOS

Para esta prueba se tomaron 15 usuarios y el umbral se fue aumentado un 0.5 %. Se tomó en cuenta los resultados de la prueba 1, para probar la hipótesis de posible no reconocimiento se sustituyeron los sujetos por otros 1, 10, 16 por 26, 29, 30 respectivamente.

### 5.2.1 prueba 2.1 umbral 0.25, 15 usuarios

Sujeto	Imágenes Reconocidas	Imágenes No Reconocidas
2	2,3,4,5,7,8,10	6,9
3	2,3,6,7,8,9,10	4,5
4	3,8,10	2,4,5,6,7,10
5	2,3,4,5,6,7	8,9,10
6	2,3,4,5,6,7,8,9,10	ninguna
7	2,3,4,10	5,6,7,8,9
11	3,6,7,10	2,4,5,8,9
12	6,8,9,10	2,3,4,5,7
13	2,3,4,5,7,8,10	3,4,6,9
14	2,3,4,5	6,7,8,9,10
15	3,4,6	2,5,7,8,9,10
17	2,5	4,6,7,8,9,10
26	2,3,4,6,7,9	5,8,10
29	2,3,4,8,9,10	5,6,7
30	2,3,6,8,10	4,5,7,9

Tabla 8: Diseño prueba 2.1.

#### Falsos Positivos:

La imagen 4,10 del sujeto 40, lo reconoce como sujeto 5.

**5.2.2 Prueba 2.2: umbral 0.30, 15 usuarios.** Se tomó la primera imagen de 15 sujetos de la base de datos de AT & T y el resto fue utilizada como prueba, el umbral fue de 0,3, los resultados fueron los siguientes:

Sujeto	Imágenes Reconocidas	Imágenes No Reconocidas
2	2,3,4,5,7,8,10	6,9
3	2,3,6,7,8,9,10	4,5
4	3,4,7,8,9,10	2,5,6
5	2,3,4,5,6,7	8,9,10
6	2,3,4,5,6,7,8,9,10	ninguna
7	2,3,4,10	5,6,7,8,9
11	3,6,7,9,10	2,4,5,8
12	2,6,8,9,10	3,4,5,7
13	2,3,4,5,6,7,8,10	9
14	2,3,4,5	6,7,8,9,10
15	3,4,6,8	2,5,7,9,10
17	2,5	3*,4,6,7,8,9,10
26	2,3,4,5,6,7,9	8,10
29	2,3,4,5,7,8,9,10	6
30	2,3,6,8,9,10	4,5,7

Tabla 9: Diseño de la prueba 2.2.

#### **Falsos Positivos:**

La imagen 3 del sujeto 18 y la imagen 4, 7, 10 del sujeto 40 lo reconoce como sujeto 5.

La imagen 3 del sujeto 17 lo reconoce como sujeto 3

La imagen 4, 6, 9 del sujeto 21, la imagen 1, 7, 8, 9 del sujeto 31 y la imagen 7,8 del sujeto 33 lo reconoce como sujeto 30.

La imagen 3,4 del sujeto 36 lo reconoce como sujeto 7.

#### **5.2.3. Análisis de resultados prueba 2.**

Prueba	Verdadero Positivo	Falso Negativo	Verdadero Negativo	Falso Positivo	Efectividad
2.1	57.03%	42.97%	99.2%	0.8%	78.11%
2.2	65.18%	6.4%	93.6%	6.4%	79.39%

Tabla 10: Resultados prueba 2.

Se observa que el porcentaje de reconocimiento individual (Verdadero Positivo) aumenta significativamente de un 22.33% con un umbral de 0.2 a un 57.03% con un umbral 0.25, esto indica que el sistema es mucho más flexible en la posición y orientación de la imagen de entrada y sensible ante cambios en el umbral de error.

También se ve por primera vez un porcentaje de falso positivo diferente de cero; 0.8%, aunque este número es muy pequeño, significa que de entre 250 imágenes de personas que no están en la base de datos, 2 fueron reconocidas como si estuvieran, violando el objetivo de sistema de control de acceso.

Cuando se aumenta el umbral de 0.25 a 0.3, se observa nuevamente un aumento en el reconocimiento individual de 57.03% a 65.18%, y un aumento considerable de la tasa de falsos positivos; de 0.8% a 6.4%. Este último porcentaje indica que de 250 imágenes de personas que no se encontraban en la base de datos, 16 fueron reconocidas como si lo estuvieran.

El sujeto de la base de datos reconocido como falso positivo en las dos pruebas es el sujeto 5. Si se analiza la imagen de dicho sujeto, se observa que está tomada a una distancia mayor, presentando

### **5.3. PRUEBA CON BASE DE DATOS REAL**

Para la siguiente prueba se tomaron 3 imágenes del rostro de 10 personas, una de ellas fue utilizada para el registros y las otras dos para pruebas. Los resultados fueron similares a las anteriores pruebas. Los usuarios tuvieron buena aceptación del sistema, el proceso de registro y reconocimiento les pareció sencillo e interesante.

## CONCLUSIONES

Con los resultados de las pruebas se puede concluir:

- Un sistema de control de acceso que utiliza el rostro para reconocimiento, tiene buena aceptación por parte de los usuarios finales, y no requiere de mucha cooperación de estos.
- A medida que el número de imágenes aumenta manteniendo un umbral constante, la capacidad de reconocer una persona disminuye.
- Los falsos positivos se presentan cuando las imágenes de entrenamiento contienen más que simplemente el rostro.
- El rendimiento del algoritmo Eigenfaces se ve afectado por la intensidad y dirección de la luz, así como la orientación del rostro del usuario.
- Con un umbral de error bajo, el sistema asegura que solo personas autorizadas ingresen al sistema, acosta de mayor restricción en la forma de la ubicación del rostro, frente al Lector (un gran parecido con la imagen en la base de datos). Con un umbral alto, el sistema es mucho más flexible para los usuarios, tolerando cierto grado de orientación del rostro, pero también, podría permitir que personas no autorizadas ingresen al sistema.
- El éxito de un sistema de control de acceso depende de la calidad de las imágenes obtenidas.
- La implantación del sistema debe incluir un proceso de afinamiento de los parámetros del sistema, que incluye el umbral de error teniendo en cuenta el número de imágenes en la base de datos, configuración del lector de rostros y adecuación de la condiciones de luz del lugar donde se instale.
- Es factible el desarrollo de un prototipo de sistema de control de acceso basado en el algoritmo Eigenfaces para permitir o no la entrada de personas a un lugar restringido.



## RECOMENDACIONES

- Tener una posición frontal del rostro, sin expresiones faciales marcadas, ni accesorios que no pertenezcan al rostro (como por ejemplo, una chaqueta) en la etapa de registro y reconocimiento.
- Acondicionar el lugar donde se vayan a realizar las pruebas.
- Implementar las funcionalidades propuestas para un sistema de control de acceso y así como otras, como por ejemplo, tener en cuenta el día y hora de ingreso permitido, diferentes lugares de ingreso con lectores de rostro y un mecanismo para registrar la salida del usuario.
- Implementar el sistema de control de acceso.
- Continuar investigando en sistemas de control de acceso.

## REFERENCIAS BIBLIOGRAFICAS

- [1] JAIN A., ROSS A., PRABHAKAR S., AN INTRODUCTION TO BIOMETRIC RECOGNITION, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 14, NO. 1, 2004
- [2] TURK M., PENTLAND A., FACE RECOGNITION USING EIGENFACES, JOURNAL OF COGNITIVE NEUROSCIENCE, PP 71-86, 1991
- [3] ZANUY F., SILVESTRE S., EXPERIMENTOS SOBRE LA VULNERABILIDAD DE SISTEMAS BIOMÉTRICOS, ESCOLA UNIVERSITÀRIA POLITÈCNICA DE MATARÓ, 2005.  
DISPONIBLE EN: [HTTP://WWW.SHSCONSULTORES.ES/JORNADAS/0203\\_-\\_EXPERIMENTOS\\_SOBRE\\_LA\\_VULNERABILIDAD\\_DE\\_SISTEMAS\\_BIOMETRICOS.PDF](http://www.shsconsultores.es/jornadas/0203_-_EXPERIMENTOS_SOBRE_LA_VULNERABILIDAD_DE_SISTEMAS_BIOMETRICOS.PDF)
- [4] SISTEMAS DE SEGURIDAD BASADOS EN CARACTERÍSTICA BIOMÉTRICA., 2003  
[HTTP://WWW.ITI.UPV.ES/FILEADMIN/CONTENIDO/01-ITI/TIC07/2005-06-BIOMETRIA.PDF](http://www.iti.upv.es/fileadmin/contenido/01-ITI/TIC07/2005-06-BIOMETRIA.PDF)
- [5] BETANCOURT A., ET AL, EFECTO DE LA INVERSIÓN DE LA IMAGEN EN EL RECONOCIMIENTO DE ROSTROS, ACTA BIOLÓGICA COLOMBIANA, VOL. 7 NO. 2, 2002.
- [6] AMAYA D., MENESES J., DETERMINACION EXPERIMENTAL DE LAS CARACTERISTICAS TRIDIMENSIONALES DEL ROSTRO CON FINES DE RECONOCIMIENTO BIOMETRICO USANDO CORRELACION TRIDIMENSIONAL, REVISTA COLOMBIANA DE FÍSICA, VOL. 38, NO. 4, 2006
- [7] INTERNATIONAL BIOMETRIC GROUP, 2009  
[HTTP://WWW.BIOMETRICGROUP.COM/REPORTS/PUBLIC/MARKET\\_REPORT.PHP](http://www.biometricgroup.com/reports/public/market_report.php).
- [8] MARCELO J. ARMENGOT IBORRA, ANÁLISIS COMPARATIVO DE MÉTODOS BASADOS EN SUBESPACIOS APLICADOS AL RECONOCIMIENTO DE CARAS, UNIVERSIDAD DE VALENCIA, 2006.
- [9] HISTORIA DE SISTEMAS DE CONTROL DE ACCESO, 2009, EN:  
[HTTP://NOTICIASINTERESANTES.BLOGCINDARIO.COM/2008/08/INDEX.HTML](http://noticiasinteresantes.blogcindario.com/2008/08/index.html)
- [10] HISTORIA DE LA BIOMETRÍA, 2009, EN  
[HTTP://RESEARCH.IIIT.AC.IN/~VANDANA/PAPERS/BASIC/BIOMETRICS\\_HISTORY.PDF](http://research.iiit.ac.in/~vandana/papers/basic/biometrics_history.pdf)

- [11] DESCRIPCIÓN DE UN SISTEMA BIOMÉTRICO, 2009, EN:  
[HTTP://WWW2.ING.PUC.CL/~IING/ED429/SISTEMAS\\_BIOMETRICOS.HTM](http://www2.ing.puc.cl/~iing/ED429/SISTEMAS_BIOMETRICOS.HTM)
- [12] RATHA N., BOLLE R. AUTOMATIC FINGERPRINT RECOGNITION SYSTEMS, SPRINGER, NEW YORK, 2004
- [13] INDIVIDUAL BIOMETRICS: IRIS SCAN, NATIONAL CENTER FOR STATE COURTS, 6 2006. DISPONIBLE EN:  
[HTTP://CTL.NCSC.DNI.US/BIOMET%20WEB/BMIRIS.HTML](http://ctl.ncsc.dni.us/biomet%20web/bmiris.html)
- [14] JHON D. WOODWARD, JR., NICHOLAS M. ORLANS, PETER T. HIGGINS, BIOMETRICS, NEW YORK: MCGRAW HILL OSBORNE, 2003.
- [15] BIGUN, J., DUC, B., SMERALDI, F., FISCHER, S., AND MAKAROV, A. MULTI-MODAL PERSON AUTHENTICATION. IN *FACE RECOGNITION: FROM THEORY TO APPLICATIONS*, H. WECHSLER, P. J. PHILLIPS, V. BRUCE, F. F. SOULIE, AND T. S. HUANG, EDS. SPRINGER-VERLAG, BERLIN, GERMANY, PP, 26–50, 1998.  
 DISPONIBLE EN [HTTP://BOOKS.GOOGLE.COM](http://books.google.com). CO ID = H1YODHFKVK8C
- [16] CRAW, I., TOCK, D., BENNETT, A., FININDING FACE FEATURES, 1987.
- [17] CRAW, I., TOCK, D., BENNETT, A., FININDING FACE FEATURES, PROCEEDINGS 2 EUROPEAN CONFERENCE ON COMPUTER VISION. 1992.  
 DISPONIBLE EN : [HTTP://EPRINTS.KFUPM.EDU.SA/41154/1/41154.PDF](http://eprints.kfupm.edu.sa/41154/1/41154.pdf)
- [18] GOVINDARAJU V. SRIHARI S., SHER D., A COMPUTATIONAL MODEL FOR FACE LOCATION, PROCEEDINGS THIRD INTERNATIONAL CONFERENCE ON COMPUTER VISION, 1990.  
 DISPONIBLE EN:  
[HTTP://WWW.AAAI.ORG/PAPERS/AAAI/1992/AAAI92-054.PDF](http://www.aaai.org/papers/aaai/1992/aaai92-054.pdf)
- [19] YUILLE A., COHEN D., HALLINAN P., FEATURE EXTRATION FROM FACES USING DEFORMABLE TEMPLATES, PROC. IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 1989.  
 DISPONIBLE EN:  
[WWW.SPRINGERLINK.COM/INDEX/TP404612X8171265.PDF](http://www.springerlink.com/index/TP404612X8171265.pdf)
- [20] BRUNELLI R., POGGIO T., FACE RECOGNITION THROUGH GEOMETRICAL FEATURES, 1992  
 DISPONIBLE EN:  
[HTTP://130.203.133.121:8080/VIEWDOC/DOWNLOAD?DOI=10.1.1.55.11&REP=REP1&TYPE=PDF](http://130.203.133.121:8080/viewdoc/download?doi=10.1.1.55.11&rep=rep1&type=pdf)

- [21] T. SAKAI, M. NAGAO, S. FUJIBAYASHI: LINE EXTRACTION AND PATTERN DETECTION IN A PHOTOGRAPH. PATTERN RECOGNITION 1(3): PP: 233-236, 1969.
- [22] REISFELD D., YESHERUN Y., ROBUST DETECTION OF FACIAL FEATURES BY GENERALIZED SYMMTRY, PROC. 11H INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION, 1992.  
DISPONIBLE EN: [HTTP://WWW.CS.TAU.AC.IL/~HEZY/PAPERS/C11.PDF](http://www.cs.tau.ac.il/~hezy/papers/c11.pdf)
- [23] BARON, R. MECHANISMS OF HUMAN FACIAL RECOGNITION, INTERNATIONAL JOURNAL OF MAN-MACHINE STUDIES, VOL 15, PP 137-178. 2002.
- [24] NIXON M., EYE SPACING MEASUREMENT FOR FACIAL RECOGNITION, SPIE PROC. 1985.  
DISPONIBLE EN:  
[HTTP://EPRINTS.ECS.SOTON.AC.UK/17382/1/NIXON\\_EYE\\_SPACING.PDF](http://eprints.ecs.soton.ac.uk/17382/1/nixon_eye_spacing.pdf)
- [25] MANJUNATH B., CHELLAPPA R., MALSBERG C., A FEATURE BASED APPROACH TO FACE RECOGNITION, PROCC. IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 1992.  
DISPONIBLE EN:  
[HTTP://VISION.ECE.UCSB.EDU/PUBLICATIONS/92CVPR.PDF](http://vision.ece.ucsb.edu/publications/92cvpr.pdf)
- [26] KAYA Y., KOBAYASHI K., A BASIC STUDY OF HUMAN FACE RECOGNITION, WATANABE, S. (ED.), FRONTIERS OF PATTERN RECOGNITION, ACADEMIC PRESS, NEW YORK, P. 265-289. 1972.
- [27] KANADE T., COMPUTER RECOGNITION OF HUMAN FACES, EN INTERDISCIPLINARY SYSTEMS RESEARCH, BIRKHAUSER VERLAG, 1977.
- [28] KIRBY M., SIROVICH L. APLICATION OF THE KARHUMEN-LOEVE PROCEDURE FOR THE CHARACTERIZATION OF HUMAN FACES, IEEE TRANS. ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE. VOL 12. NO 1 JANUARY; PP. 103-108. 1990.  
DISPONIBLE EN:  
[HTTP://WWW.CSEE.WVU.EDU/~NATALIAS/BIOM426/KIRBY90.PDF](http://www.csee.wvu.edu/~natalias/biom426/kirby90.pdf)
- [29] JAIN A. FUNDAMENTALS OF DIGITAL IMAGE PROCESSING, PRENTICE-HALL, ENGLEWOOD, 1989.
- [30] R., POGGIO T., AUTOMATIC PERSON RECOGNITION BY ACOUSTIC AND GEOMETRIC FEATURES, MVA, V. 8, P. 317-325, 1995.

- [31] BRUNELLI R., POGGIO T., FACE RECOGNITION: FEATURES VERSUS TEMPLATES, IEEE TRANS. ON PATTERN RECOGNITION AND MACHINE INTELLIGENCE V.15, NO. 10, OCTOBER. PP. 1042-1052,1993.
- [32] LUCAS S., THE CONTINUOUS N-TUPLE CLASSIFIER AND ITS APPLICATIONS TO FACE RECOGNITION. 1994.  
DISPONIBLE EN:  
[HTTP://CITSEERX.IST.PSU.EDU/VIEWDOC/DOWNLOAD?DOI=10.1.1.57.1793  
&REP=REP1&TYPE=PDF](http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.57.1793&rep=rep1&type=pdf)
- [33] KOHONEN T., SELF-ORGANIZATION AND ASSOCIATIVE MEMORY, SPRINGER-VERLAG,BERLIN,1988.
- [34] STONHAM T., PRACTICAL FACE RECOGNITION AND VERIFICATION WIDH WISARD, EN ELLIS H., JEEVES M., NEWCOMBE F., ASPECTS OF FACE PROCESSING, DORDRECHT, NIJHOFF, 1986.
- [35] BRUNELLI R., POGGIO T., HYPERBF NETWORKS FOR GENDER CLASSIFICATION, PROCEEDING DARPA IMAGE UNDERSTANDING WORKSHOP, PP 311-314, 1992.
- [36] FUKUNAGA K., STATISTICAL PATTERN RECOGNITION, ACADEMIC PRESS, NEW YORK, 1989.
- [37] SEIBERT M., WAXMAN A., RECOGNIZIN FACES FROM THEIR PARTS, SPIE PROC., V.1611: SENSOR FUSION IV-CONTROL PARADIGMS AND DATA STRUCTURES, PP. 129-140. 1991.
- [38] LIU C., WECHSLER H., FACE RECOGNITION USING EVOLUTIONARY PURSUIT, FFTH EUROPEAN CONFERENCE ON COMPUTER VISION, UNIVERSITY OF FREIBURG, GERMANY, 1998.
- [39] BHANU B., LEE S., MING J., ADAPTIVE IMAGE SEGMENTATION USING A GENETIC ALGORITHM, IEEE TRANS. ON SYSTEMS, MAN, AND CYBERNETICS, VOL 25, NO. 12, DECEMBER; PP. 1543-1567, 1995.
- [40] HUANG R., DETECTION STRATEGIES FOR FACE RECOGNITION USING LEARNING AND EVOLUTION, PH.D. DISSERTATION, GEORGE MASON UNIVERSITY, 1998.
- [41] PINTO R., SOSSA J., HUMAN FACE IDENTIFICATION USING INVARIANT DESCRIPTIONS AND GENETIC ALGORITHM, EN COELHO H., PROGRESS IN ARTIFICIAL INTELLIGENCE-IBERAMIA, 1998.
- [42] JACOBI ROTATION, 2008 EN:

[HTTP://BEIGE.UCS.INDIANA.EDU/B673/NODE24.HTML](http://beige.ucs.indiana.edu/B673/node24.html)

[43] THE HOUSEHOLDER-QR/QL ALGORITHM, 2008 EN:  
[HTTP://BEIGE.UCS.INDIANA.EDU/B673/NODE30.HTML](http://beige.ucs.indiana.edu/B673/node30.html)

[44] PERSON K., ON LINES AND PLANES OF CLOSEST FIT TO SYSTEM POINT IN THE SPACE. PHILOSOPHICAL MAGAZINE, VOL 6, PP 559-572, 1991.

[45] HOTTELING H., ANALISIS OF A COMPLEX OF STATICAL VARIABLES INTO PRINCIPAL COMPONENTS. JOURNAL OF EDUCATIONAL PSYCHOLOGY, 24, PP 417-441; 498-520. 1933.

[46] THURSTONE L., MULTIPLE FACTOR ANALYSIS. CHICAGO. UNIVERSITY OF CHICAGO PRESS, 1997.

[47] KAISER H., THE VARIMAX CRITERION FOR ANALYTIC ROTATION IN FACTOR ANALYSIS. PSYCHOMETRIKA, 23, PP 187-200. 1958.

[48] FUKUNAGA K., INTRODUCTION TO STATISTICAL PATTERN RECOGNITION, ACADEMIC PRESS, NEW YORK, 1972.

[49] DAVID R. HILL, BERNARD KOLMAN, ÁLGEBRA LINEAL, PAG 433 – 442.  
DISPONIBLE EN [HTTP://BOOKS.GOOGLE.COM.CO/ ID=VO9AWRASL74C](http://books.google.com.co/id=VO9AWRASL74C)

[50] RUP, 2008, EN:  
[HTTP://ES.WIKIPEDIA.ORG/WIKI/PROCESO\\_UNIFICADO\\_DE\\_RATIONAL](http://es.wikipedia.org/wiki/Proceso_unificado_de_rational)

## **OTRAS REFERENCIAS**

HOLSEHOLDER REDUCTION, 2008, EN:  
[HTTP://BEIGE.UCS.INDIANA.EDU/B673/NODE31.HTML](http://beige.ucs.indiana.edu/B673/node31.html)

GRENANDER U., CHOW Y., KEENAN, D., HANDS: A PATTERN THEORETIC STUDY OF BIOLOGICAL SHAPES, SPRINGER-VERLANG, NEW YORK, 1991.

BRUNELLI R., POGGIO T., CARICATURAL EFFECT IN ATOMATED FACE PERCEPTION, 1990.

THURSTONE L., THE VECTORS OF THE MIND. CHICAGO. UNIVERSITY OF CHICAGO PRESS, 1935.

INFORMACIÓN GENERAL Y CONCEPTUAL SOBRE .NET FRAMEWORK, 2009

EN: [HTTP://MSDN.MICROSOFT.COM/ES-ES/LIBRARY/ZW4W595W.ASPX](http://msdn.microsoft.com/es-es/library/zw4w595w.aspx)

INTRODUCCIÓN A WPF, 2009

EN: [HTTP://MSDN.MICROSOFT.COM/ES-ES/LIBRARY/AA970268.ASPX](http://msdn.microsoft.com/es-es/library/aa970268.aspx)

INTRODUCCIÓN A WCF, 2009 EN:

[HTTP://MSDN.MICROSOFT.COM/ES-ES/LIBRARY/MS731082.ASPX](http://msdn.microsoft.com/es-es/library/ms731082.aspx)

CONCEPTOS BÁSICOS DE WCF, 2009 EN:

[HTTP://MSDN.MICROSOFT.COM/ES-ES/LIBRARY/MS731079.ASPX](http://msdn.microsoft.com/es-es/library/ms731079.aspx)

MICROSOFT VISUAL ESTUDIO, 2009 EN:

[HTTP://ES.WIKIPEDIA.ORG/WIKI/MICROSOFT\\_VISUAL\\_STUDIO](http://es.wikipedia.org/wiki/Microsoft_Visual_Studio)

EXPRESSION STUDIO, 2009

EN: [HTTP://ES.WIKIPEDIA.ORG/WIKI/EXPRESSION\\_STUDIO](http://es.wikipedia.org/wiki/Expression_Studio)

BASE DE ROSTROS DE AT & T, 2008

EN:

[HTTP://WWW.CL.CAM.AC.UK/RESEARCH/DTG/ATTARCHIVE/FACEDATABASE.HTML](http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html)

PARKER R., ALGORITHMS FOR IMAGE PROCESSING AND COMPUTER VISION, JHON WILEY & SONS, 1997.

DILLON W., GOLDSTEIN M., MULTIVARIATE ANALYSIS: METHODS AND APPLICATIONS. JHON WILEY & SONS, 1984

EJEMPLO DE WPF 2009.

EN: [HTTP://BLACKOUT360.WORDPRESS.COM/2008/06/02/EJEMPLO-DE-VARIAS-COSAS-CON-WPF-CODIGO-FUENTE/](http://blackout360.wordpress.com/2008/06/02/ejemplo-de-varias-cosas-con-wpf-codigo-fuente/)

RITTER G., WILSON J. HANDBOOK OF COMPUTER VISION ALGORITHMS IN IMAGE ALGEBRA, SECOND EDITION, CRC PRESS, 2000,

BALLAR D., BROWN C. COMPUTER VISION.

VILLEGAS C., RECONOCIMIENTO DE ROSTROS UTILIZANDO ANÁLISIS DE COMPONENTES PRINCIPALES: LIMITACIONES DEL ALGORITMO, UNIVERSIDAD IBEROAMERICANA, 2004.

ZHAO W., CHELLAPPA R., ROSENFELD A., FACE RECOGNITION: A LITERATURE SUERVEY, UNIVERSITY OF MARYLAND, 2006.

CANAVOS G., PROBABILIDAD Y ESTADÍSTICA, APLICACIONES Y MÉTODOS, VIRGINIA COMMONWEALTH UNIVERSITY, 1998.

VILLA S., SISTEMA DE RECONOCIMIENTO DE ROSTROS, TRABAJO DE PREGRADO, UNIVERSIDAD PERUANA DE CIENCIAS APLICADAS, 2007.

SHLENS J., A TUTORIAL ON PRINCIPAL COMPONENT ANALYSIS, CENTER FOR NEURAL SCIENCE, NEW YORK UNIVERSITY, 2002.

PRESS W., TEUKOISY S., NUMERICAL RECIPES, THE ART OF SCIENTIFIC COMPUTING, CAMBRIDGE UNIVERSITY PRESS, 2001, .

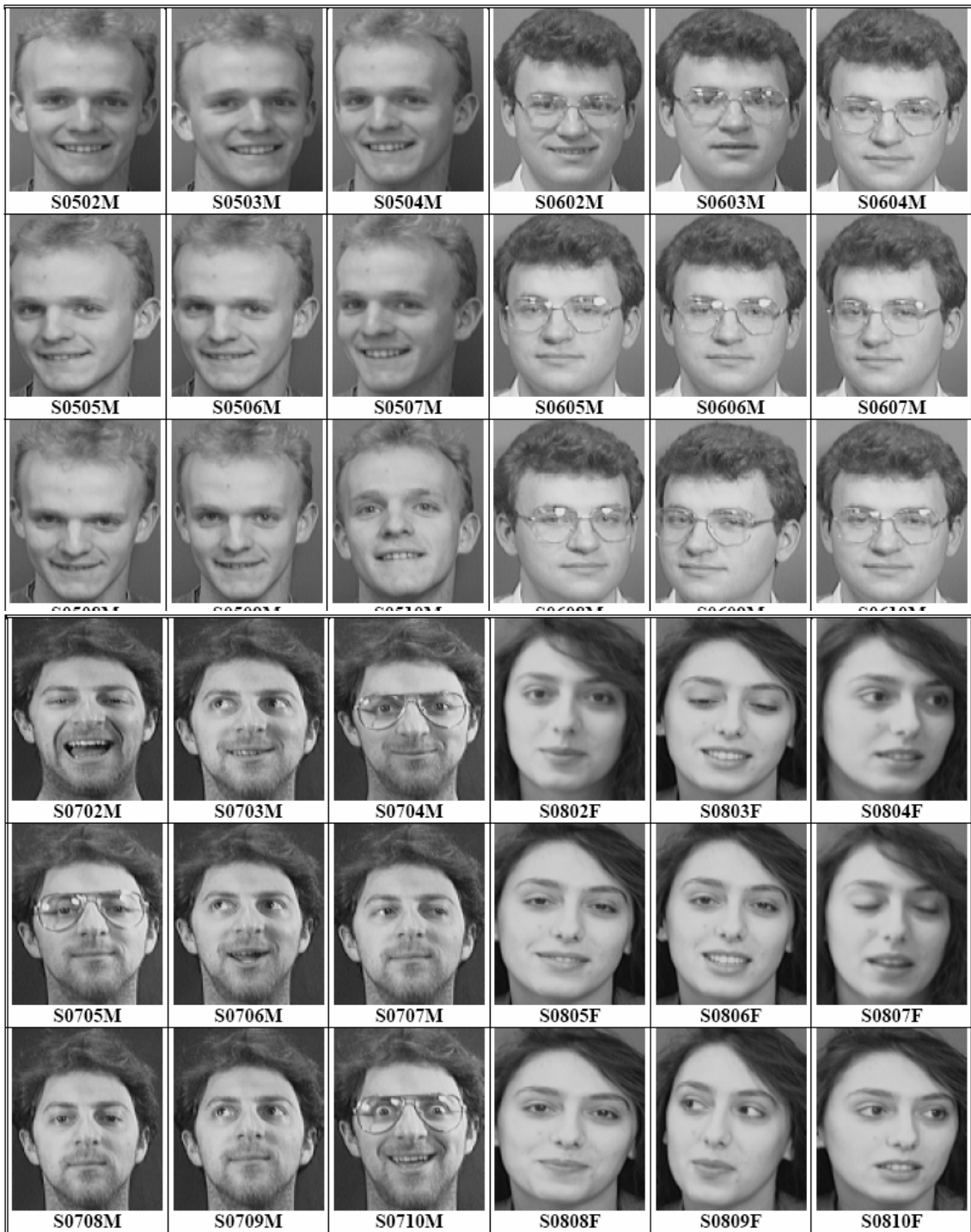
CAIFA C., DESARROLLO DE UN SOFTWARE PARA LA IDENTIFICACIÓN DE ELEFANTES MARINOS POR EIGENFACES, LAB SISTEMAS COMPLEJOS, UNIVERSIDAD DE CHILE, 2005.

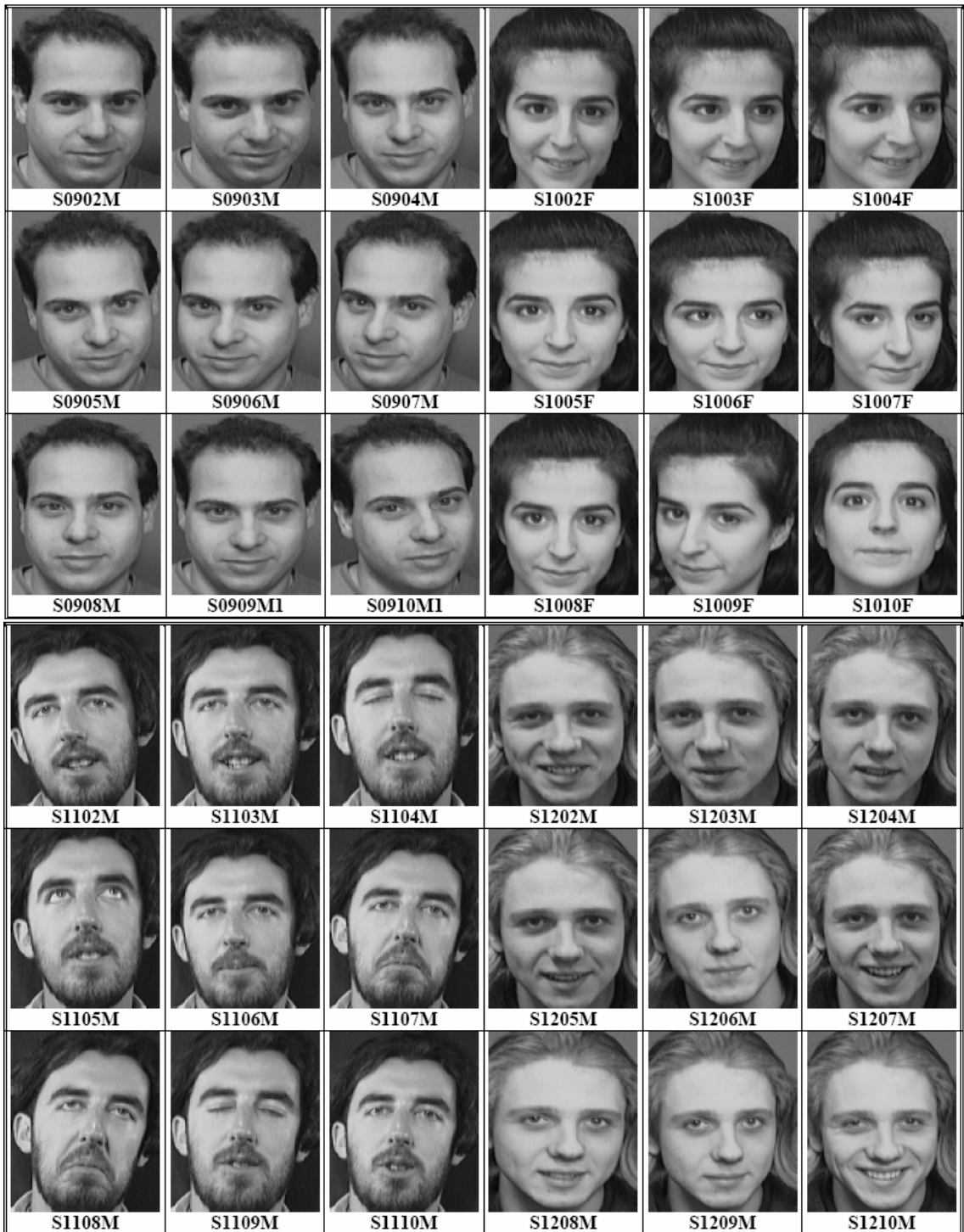


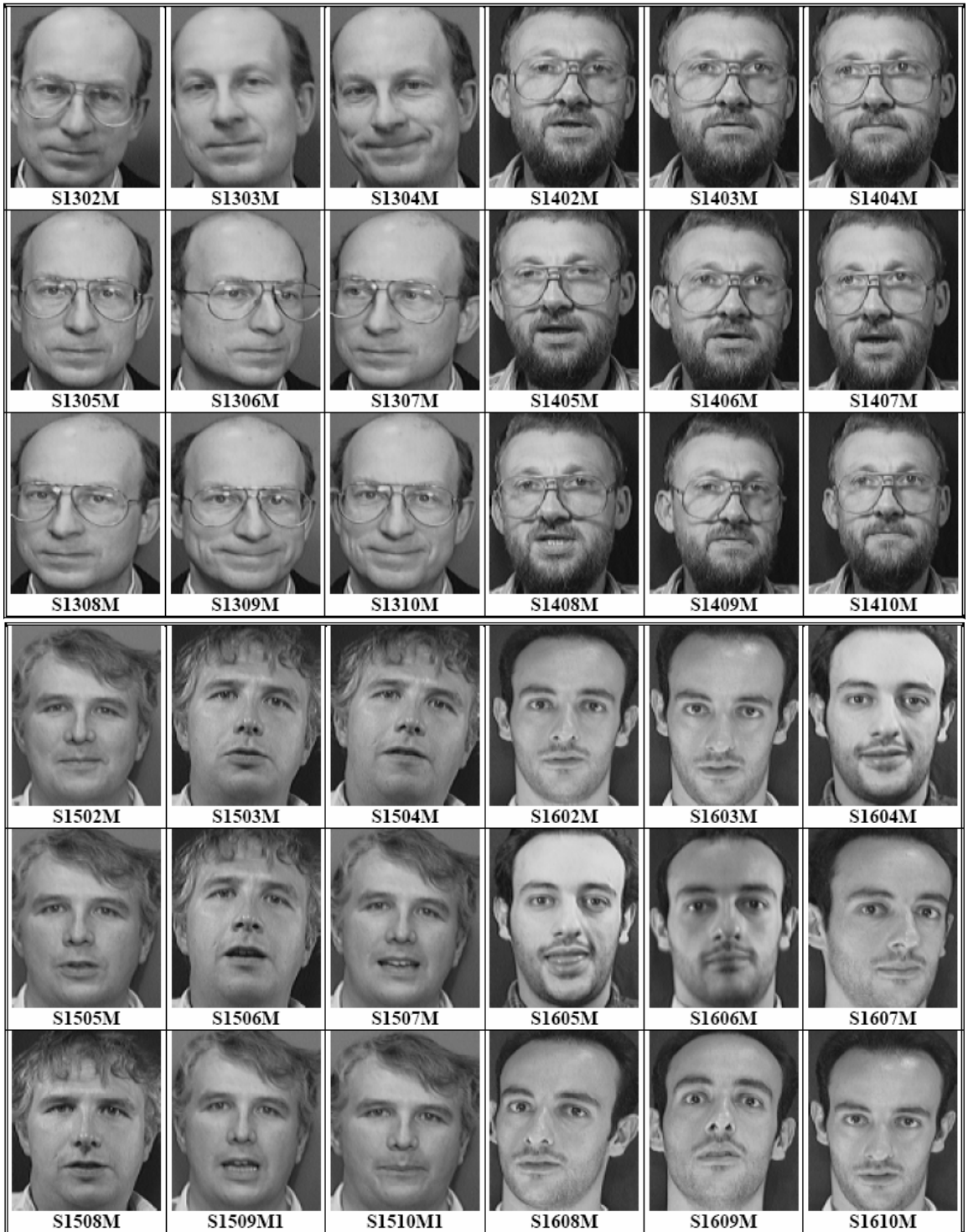
# ANEXOS

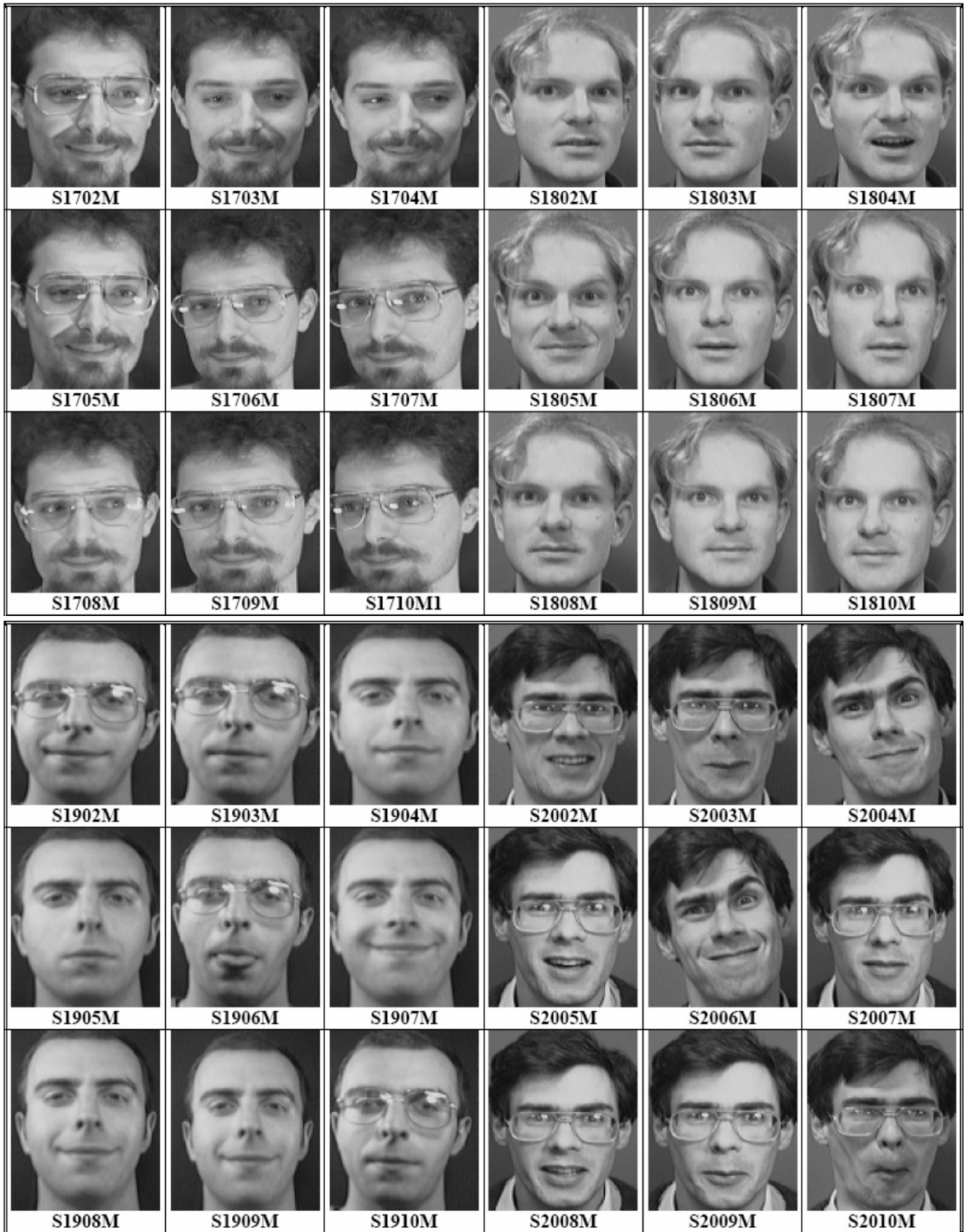
## Anexo 1. BASE DE ROSTROS AT & T























## Anexo 2. BASE DE ROSTROS REAL

