

**PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA**

**DESARROLLO DE UN AUTENTICADOR CON KINECT MEDIANTE
RECONOCIMIENTO FACIAL**

ANGELO NICOLÁS ASTE TOLEDO

SEBASTIÁN RAMÓN PRADO CABRERA

**INFORME FINAL DEL PROYECTO
PARA OPTAR AL TÍTULO PROFESIONAL DE
INGENIERO DE EJECUCIÓN EN INFORMÁTICA**

AGOSTO 2015

Pontificia Universidad Católica de Valparaíso
Facultad de Ingeniería
Escuela de Ingeniería Informática

**DESARROLLO DE UN AUTENTICADOR CON KINECT MEDIANTE
RECONOCIMIENTO FACIAL**

ANGELO NICOLÁS ASTE TOLEDO

SEBASTIÁN RAMÓN PRADO CABRERA

Profesor Guía: **José Miguel Rubio León.**

Profesor Co-referente: **Rodrigo Alfaro Arancibia.**

Carrera: **Ingeniería de Ejecución en Informática**

Agosto 2015

Agradecimientos

A todas las personas que han estado conmigo y me apoyaron desde mis inicios, en especial a mi familia y seres queridos, que siempre se preocuparon por mí, entregándome todo su cariño. Ustedes me han enseñado lo más importante de esta vida, el amor. Siempre estaré agradecido de todo lo que me han entregado. ¡Muchas Gracias!

Sebastián Prado

Agradecimientos

A mi familia y a mis seres queridos por comprenderme, darme su cariño y apoyarme en estos 4 años de carrera.

A mis amigos y compañeros de la universidad por todos las salidas, las aventuras y el camino que hemos recorrido.

A mi profesor guía José Miguel, por ayudarnos, orientarnos y aclarar mis ideas a veces dispersas en este proyecto.

¡Muchas Gracias!

Angelo Aste

Resumen.

A medida que el mundo evoluciona la humanidad, se ha visto en la necesidad de realizar avances en la seguridad. La identificación facial es uno de los temas más estudiados, debido a las ventajas que posee frente a otros métodos de identificación. En el siguiente documento se muestra el trabajo realizado para este proyecto de titulación, que consistió en la implementación de un sistema autenticador facial basado en puntos característicos, empleando el dispositivo Kinect para la adquisición de las imágenes a través de la API ofrecida por Microsoft. Se compararon diferentes técnicas existentes que favorecían el reconocimiento facial con este dispositivo.

Esto permitió mostrar que el uso de cálculo de distancias y áreas aplicadas en imágenes de profundidad tiene un mejor desempeño para lograr la autenticación de personas frente a otros métodos que fueron evaluados en esta investigación.

Abstract

As the world evolves, the humanity has seen the need of making advances in the security area. The facial identification is one of the most studied topics because of the advantages that it has in comparison to other identification methods. In this document, it is shown the work done for this degree's project, which consists in the implementation of a facial authenticator system based on characteristic points, using the Kinect device for the images obtaining through the API provided by Microsoft. Different techniques that help with the facial recognition with this device were compared.

This allowed to show that calculating distances and areas, applied to images in deep, has a better performance to achieve the authentication of people in comparison to other methods that were evaluated in this research.

Índice

Resumen.....	i
Abstract.....	i
Índice.....	ii
Lista de Figuras.....	vi
Lista de Tablas.....	viii
1 Introducción.....	1
1.1 Objetivos.....	2
1.1.1 Objetivo General.....	2
1.1.2 Objetivos Específicos.....	2
1.2 Metodología de desarrollo.....	3
1.3 Planificación.....	4
1.3.1 Plan de trabajo tentativo.....	4
2 Marco Teórico.....	6
2.1 Tecnología Kinect.....	6
2.1.1 Definición.....	6
2.1.2 Características.....	6
2.1.3 Especificaciones Técnicas.....	7
2.1.4 Usos del dispositivo Kinect.....	8
2.2 Tecnologías existentes con reconocimiento facial.....	9
2.3 Pasos generales en un reconocimiento facial.....	9
2.3.1 Adquisición de la imagen.....	10
2.3.2 Pre-procesamiento de la imagen.....	10
2.3.3 Localización de la cara.....	10
2.3.4 Extracción de las características faciales.....	10
2.3.5 Reconocimiento Facial.....	11
2.4 Técnicas de reconocimiento facial.....	11
3 Requerimientos.....	12
3.1 Requerimientos Funcionales.....	12
3.2 Requerimientos No Funcionales.....	12
4 Estudio de Factibilidad.....	13
4.1 Factibilidad técnica.....	13

4.1.1	A nivel de hardware	13
4.1.2	A nivel de software	13
4.1.3	A nivel personal	13
4.2	Factibilidad Económica	13
4.3	Factibilidad Operativa.....	13
4.4	Factibilidad Legal	14
5	Diseño de la Solución.....	15
5.1	Arquitectura de la solución.....	15
5.2	C Sharp (C#).....	16
5.3	Interfaz de Seguimiento Facial de Kinect (Face Tracking)	16
5.3.1	Descripción de la API	18
5.4	¿Por qué Kinect?.....	19
5.5	Técnica Utilizada	19
5.6	Casos de Uso.....	22
5.6.1	Autenticar Usuario	22
5.6.2	Registrar Usuarios.....	23
5.6.3	Captura del rostro	24
5.6.4	Listar usuarios	24
5.6.5	Editar usuario	25
5.6.6	Eliminar usuario	25
5.6.7	Buscar usuario	26
5.7	Diagrama de Clases	27
5.8	Diagramas de Secuencia	27
5.8.1	Registrar Usuario.....	28
5.8.2	Autenticar Usuario	28
5.8.3	Editar usuario	29
5.9	Diagramas de actividad.....	30
5.9.1	Registro de usuario.....	30
5.9.2	Autenticar Usuario	31
5.10	Modelo Relacional.....	32
6	Implementación de la solución.....	33
6.1	Prototipo Final	33
6.1.1	Menú.....	33

6.1.2	Inicio de reconocimiento facial	33
6.1.3	Registrar usuario	35
6.1.4	Listar usuarios	37
6.1.5	Detalle Usuario.....	37
6.2	Problemas y limitaciones observadas	38
6.3	Resultados de identificación obtenidos.....	40
6.3.1	Distancias	40
6.3.2	Áreas.....	41
6.3.3	Distancias Normalizando Datos	41
6.3.4	Áreas Normalizando Datos	42
6.3.5	Distancias Aplicando PCA.....	42
6.3.6	Áreas Aplicando PCA.....	42
6.3.7	Promedio Usando Distancias	43
6.3.8	Promedio Usando Áreas.....	43
6.3.9	Promedio Usando Áreas y Distancias	44
7	Conclusiones	45
8	Referencias	46
9	Anexos.....	47
9.1	Conceptos.....	47
9.2	Principales Técnicas Holísticas	48
9.2.1	Principal Component Analysis (PCA)	48
9.2.2	Análisis de discriminante lineal (LDA)	48
9.2.3	Análisis de Componentes Independientes (ICA)	49
9.2.4	Métodos Kernel	49
9.2.5	Evolutionary Pursuit (EP)	50
9.2.6	Suport Vector Machine (SVM).....	50
9.3	Principales Técnicas Basadas en Rasgos Locales.....	51
9.3.1	AAM Active Appearance Model	51
9.3.2	Elastic Bunch Graph Matching (EBGM)	52
9.3.3	Modelos de Markov escondidos (HMM)	55
9.3.4	Local Binary Patterns (LVP).....	56
9.3.5	Modelado 3D.....	57
9.4	Descripción de la API	60

9.4.1	IFTFaceTracking.....	60
9.4.2	IFTResult.....	62
9.4.3	IFTImage.....	62
9.4.4	IFTModel	62
9.5	Algoritmo para la extracción de coordenadas.....	63
9.6	Resultados.....	63
9.6.1	Problemas y limitaciones observadas.....	63

Lista de Figuras

Figura 1-1 Modelo Evolutivo [15].....	3
Figura 2-1 Sensor Kinect de Microsoft [11]	6
Figura 2-2 Rango Válido del Sensor de Profundidad	7
Figura 2-3 Localización y segmentación con Viola-Jones [12].....	10
Figura 5-1 Interacción con la aplicación [11]	15
Figura 5-2 Estructura de la solución	16
Figura 5-3 FaceTracking	17
Figura 5-4 Comparación FaceTracking	17
Figura 5-5 Casos de uso	22
Figura 5-6 Diagrama de Clases	27
Figura 5-7 Diagrama de secuencia Registro de usuario	28
Figura 5-8 Diagrama de secuencia Autenticar usuario	28
Figura 5-9 Diagrama de secuencia Editar usuario	29
Figura 5-10 Diagrama de actividad Registro de usuario.....	30
Figura 5-11 Diagrama de actividad Autenticar Usuario	31
Figura 5-12 Modelo Relacional	32
Figura 6-1 Menú.....	33
Figura 6-2 Inicio de reconocimiento	34
Figura 6-3 Proceso de captura.....	34
Figura 6-4 Reconocimiento exitoso	35
Figura 6-5 Formulario de registro	35
Figura 6-6 Captura del rostro	36
Figura 6-7 Captura exitosa.....	36
Figura 6-8 Listar usuarios	37
Figura 6-9 Detalle usuario.....	37
Figura 6-10 Seguimiento del rostro	38
Figura 6-11 Puntos considerados	39
Figura A9-1 PCA [5].....	48
Figura A9-2 LDA [5]	49
Figura A9-3 SVM [10].....	50
Figura A9-4 Cara con Landmarks [9].....	51

Figura A9-5 Partición de la textura en zonas [9]	52
Figura A9-6 Representación de la cara en forma de grafo [6]	53
Figura A9-7 Representación de un FBG [6]	53
Figura A9-8 Ejemplo de grafos obtenidos utilizando EBGM [6]	54
Figura A9-9 A la izquierda, división en franjas de un rostro para el algoritmo HMM. A la derecha, las cadenas de nivel inferior y superior para el algoritmo HMM [8]	56
Figura A9-10 LVP [16]	56
Figura A9-11 Para el reconocimiento, los coeficientes α y β de la imagen de prueba, se comparan con los coeficientes de la colección de imágenes de entrenamiento [4]	58
Figura A9-12 Modelo de Cara representado por una nube de puntos [14]	59
Figura A9-13 La red modelada en diferentes niveles. (a) red Básica. (b) de Primer Nivel (c) de Segundo Nivel. (d) de Tercer Nivel. (e) de Cuarto Nivel [14]	59

Lista de Tablas

Tabla 1 Plan de trabajo.....	5
Tabla 2 Interfaces de la API.....	18
Tabla 3 Caso de uso Autenticar Usuario.....	23
Tabla 4 Caso de uso Registrar Usuarios.....	23
Tabla 5 Caso de uso Capturar rostro.....	24
Tabla 6 Caso de uso Listar Usuarios.....	25
Tabla 7 Caso de uso Editar Usuario.....	25
Tabla 8 Caso de uso Eliminar Usuario.....	26
Tabla 9 Caso de uso Buscar usuario.....	26
Tabla 10 Resultado comparación distancias.....	39
Tabla 11 Distancias.....	40
Tabla 12 Áreas.....	41
Tabla 13 Distancias Normalizando Datos.....	42
Tabla 14 Áreas Normalizando Datos.....	42
Tabla 15 Distancias Aplicando PCA.....	42
Tabla 16 Areas Aplicando PCA.....	43
Tabla 17 Promedio Usando Distancias.....	43
Tabla 18 Promedio Usando Áreas.....	44
Tabla 19 Promedio Usando Áreas y Distancias.....	44
Tabla 20 Alumno 2 (1.2 mts).....	64
Tabla 21 Alumno 1 (1.2 mts).....	64
Tabla 22 Alumno 1 (1.5 mts).....	65
Tabla 23 Alumno 2 (1.5 mts).....	65
Tabla 24 Alumno 1 (1.65 mts).....	66
Tabla 25 Alumno 2 (1.65 mts).....	66
Tabla 26 Alumno 1 (1.8 mts).....	67
Tabla 27 Alumno 2 (1.8mts).....	67

1 Introducción

En la actualidad, la seguridad es un factor muy importante tanto en la vida como en una empresa, convirtiéndose en un tema activo de investigación. Dentro de la seguridad, la autenticación de usuarios es uno de los elementos clave que ha mantenido un desarrollo constante, surgiendo novedosas formas para autenticar, ofreciendo altos niveles de seguridad, con el fin de lograr una correcta identificación de la persona y así evitar problemas de suplantación.

Hoy en día, la autenticación biométrica, y en particular el reconocimiento facial es un tema que ha ido en alza, debido a la gran satisfacción que entrega a las personas, por ser un método no intrusivo, que utiliza el rasgo anatómico más utilizado por los seres humanos para identificar a otro individuo. Es por esta razón que cada vez existen más campos de aplicaciones de esta tecnología, utilizando el rostro como forma de identificación, evitando el uso de tarjetas o de claves que pueden ser sustraídas a los usuarios. Aunque los humanos reconocemos caras familiares con poco o ningún esfuerzo, la automatización de esta tarea para ser llevada a cabo, es enormemente compleja. La causa principal de la enorme dificultad que entraña este problema viene dada por las enormes variaciones existentes entre las distintas imágenes posibles de un mismo individuo.

El trabajo que se realizó consiste en detectar el rostro humano a través de un dispositivo llamado Kinect, y así utilizar su tecnología para autenticar personas en tiempo real. Para esto se realizó una investigación sobre estas nuevas tecnologías y las técnicas que se pueden utilizar para lograr este propósito.

1.1 Objetivos

1.1.1 Objetivo General

Desarrollo de una aplicación para la identificación de personas, basada en la tecnología Kinect.

1.1.2 Objetivos Específicos

- Identificar las capacidades del dispositivo Kinect y las técnicas que favorezcan el reconocimiento facial, permitiendo una mayor tolerancia a factores externos.
- Lograr la detección y reconocimiento de rostros a través del Face tracking y guardar capturas 3D en una Base de datos.
- Lograr saber que distancia es la más adecuada para realizar una identificación de un rostro utilizando el dispositivo Kinect. Alcanzando una tasa de efectividad al menos del 80%.

1.2 Metodología de desarrollo

La metodología de desarrollo seleccionada para este proyecto es evolutivo basado en prototipos. Esa metodología se centra en la idea de ayudar a comprender los requisitos del sistema sobre todo si no se tiene una idea clara de lo que se desea. Además se puede utilizar cuando existen dudas de la viabilidad de la solución pensada. A medida que avanza el proyecto se pueden presentar prototipos al cliente de la parte ya desarrollada del proyecto y se continúa el desarrollo del prototipo en base a la retroalimentación recibida. Este proceso itera hasta que se llegue al producto final.

El uso de un modelo evolutivo basado en prototipos hace mucho más flexible el desarrollo del software, ya que permite la opción de realizar cambios en etapas tempranas. Además de dar la posibilidad de emitir varios prototipos durante el desarrollo, viendo aumentada su funcionalidad con cada avance [15].

Usando esta metodología, las etapas de ciclo de vida clásico se pueden modificar de la siguiente manera.

- Análisis de los requerimientos de sistema.
- Diseño, desarrollo e implementación del prototipo.
- Prueba y evaluación del prototipo.
- Refinamiento del prototipo y sus especificaciones.
- Diseño e implementación del sistema final.
- Explotación y mantenimiento.

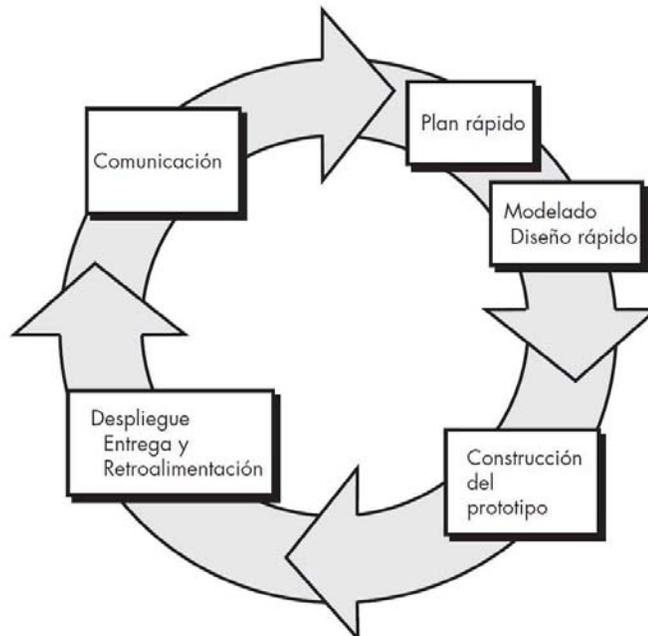


Figura 1-1 Modelo Evolutivo [15]

El principal inconveniente es que se desconoce el tiempo que se tardará en crear un producto aceptable. No se sabe cuántas iteraciones se tendrán que realizar. Otro inconveniente es que se pueden adoptar prácticas de programación de prueba-y-error, sin un análisis y diseño formales previos.

1.3 Planificación

1.3.1 Plan de trabajo tentativo

Nombre de la Tarea	Duración	Comienzo	Fin
Proyecto 1			
Análisis Preliminar de los requerimientos			
Definir ámbito, alcance y objetivos del proyecto	10 días	26.07.2014	05.08.2014
Desarrollo del marco teórico sobre tecnología Kinect y conceptos.	11 días	11.08.2014	16.08.2014
Investigación de distintos usos del dispositivo Kinect	6 días	17.08.2014	21.08.2014
Investigación de las distintas técnicas usadas para reconocimiento facial	5 días	22.08.2014	28.08.2014
Desarrollo del marco teórico sobre técnicas de reconocimiento facial.	7 días	29.09.2014	04.09.2014
Investigación de sistemas con reconocimiento facial.	6 días	01.08.2014	05.08.2014
Estudio de Factibilidad	5 días	06.09.2014	08.09.2014
Estudio de los requerimientos del sistema	3 días	09.09.2014	12.09.2014
Diseño Rápido del sistema			
Desarrollo de Análisis de la solución	4 días	13.09.2014	17.09.2014
Desarrollo de casos de uso	5 días	18.09.2014	20.09.2014
Desarrollo Informe Avance Proyecto 1	5 días	21.09.2014	26.09.201
Investigación de entorno de desarrollo	10 días	01.10.2014	10.10.2014
Investigación sobre Face tracking y lenguaje C#	15 días	11.10.2014	25.10.2014
Modelado UML	7 días	26.10.2014	01.11.2014
Construcción del Prototipo Versión inicial			
Desarrollo de prototipo 1	13 días	02.11.2014	14.11.2014
Desarrollo Informe Proyecto 1	7 días	15.11.2014	21.11.2014
Evaluación del prototipo			
Presentación proyecto 1	1 día	02.12.2014	02.12.2014
Proyecto 2			
Refinamiento del prototipo y sus especificaciones.			
Refinamiento requerimientos	5 días	02.03.2015	06.03.2015
Estudio sobre limitaciones del dispositivo Kinect	7 días	07.03.2015	13.03.2015
Diseño Rápido del sistema			
Hacer pruebas sobre la distancia Kinet – rostro más eficiente	5 días	14.03.2015	18.03.2015
Identificación de puntos más	5 días	19.03.2015	23.03.2015

significativos.			
Construcción del Prototipo			
Implementación prototipo con distancias y áreas de puntos más significativos	10 días	24.03.2015	03.04.2015
Evaluación del prototipo			
Hacer pruebas de autenticación y evaluación de resultados obtenidos	2 días	04.04.2015	05.04.2015
Refinamiento del prototipo y sus especificaciones.			
Cambios en los umbrales definidos	2 días	06.04.2015	07.04.2015
Diseño y Construcción del Prototipo			
Implementación de cambios en los umbrales	4 días	08.04.2015	11.04.2015
Evaluación del prototipo			
Hacer pruebas de autenticación y evaluación de resultados obtenidos	2 días	12.04.2015	13.04.2015
Refinamiento umbrales	2 días	12.04.2015	13.04.2015
Diseño y Construcción del Prototipo			
Investigación Normalización y reconocimientos de objetos 3D	6 días	13.04.2015	17.04.2015
Implementación de prototipo aplicando Normalización	8 días	18.04.2015	25.04.2015
Evaluación del prototipo			
Hacer pruebas de autenticación y evaluación de resultados obtenidos	5 días	26.04.2015	30.04.2015
Presentación de avance proyecto 2	1 día	30.04.2015	30.04.2015
Refinamiento del prototipo y sus especificaciones.			
Replanteo de la solución para mitigar dispersión de los datos	4 días	01.05.2015	04.05.2015
Diseño rápido			
Modelamiento y pseudocódigo de nueva solución	8 días	05.05.2015	12.05.2015
Construcción del Prototipo			
Desarrollo de prototipo	20 días	13.05.2015	01.06.2015
Desarrollo Informe Proyecto 2	7 días	02.06.2015	08.06.2015
Evaluación del prototipo por el cliente			
Presentación proyecto 2	1 días	09.06.2015	09.06.2015
Refinamiento del prototipo			
Ajuste en los umbrales definidos	4 días	10.06.2015	14.06.2015
Plan de pruebas	10 días	11.06.2015	20.06.2015
Producto de ingeniería			
Desarrollo sistema final	35 días	21.06.2015	25.07.2015

Tabla 1 Plan de trabajo

2 Marco Teórico

2.1 Tecnología Kinect

2.1.1 Definición

Kinect es un controlador de juego libre creado por Alex Kipman, y desarrollado por Microsoft, originalmente diseñado para la consola de videojuegos Xbox 360, con el propósito de que los jugadores pudiesen interactuar con la consola de una manera más natural e intuitiva sin tener un contacto físico con el control tradicional de la consola, de manera que la jugabilidad fuese mucho más ágil y los juegos no requiriesen aprendizaje por parte de los usuarios para poder dominarlos [11].

A pesar de que en un principio el dispositivo Kinect fue pensado para ser usado en videojuegos, el 12 de junio del 2011 fue lanzada la “Kinect for Windows”, lanzando un kit de Desarrollo (SDK) para poder dar la oportunidad a los desarrolladores de crear nuevas aplicaciones que usen el dispositivo en PCs de manera oficial, ajenas al uso de la Xbox 360.

2.1.2 Características

Este dispositivo consta de cuatro elementos principales mostrados en la figura 2-1: una cámara RGB, sensor de profundidad (proyector IR + cámara IR), un motor para controlar la inclinación del dispositivo y un arreglo de cuatro micrófonos distribuidos a lo largo del sensor. Gracias a estos, este dispositivo puede reconocer gestos, comandos de voz, objetos, e imágenes, posicionándolos en un sistema de ejes X, Y, Z, siendo capaz de identificar una serie de puntos del cuerpo humano [11].

Gracias a sus características, esta tecnología ha llamado la atención de las demás áreas, y se ha pensado en incluir esta tecnología en áreas como la salud, aprendizaje, industria, seguridad, entre otras.



Figura 2-1 Sensor Kinect de Microsoft [11]

2.1.3 Especificaciones Técnicas

El sensor de profundidad permite tener una visión 3D del entorno del dispositivo Kinect. El emisor genera rayos infrarrojos y el sensor detecta los rayos que rebotan en los objetos o personas de la habitación. Los rayos reflejados son convertidos en información de profundidad para poder medir la distancia entre un objeto y el sensor, lo que permite generar la imagen de profundidad. Esto es especialmente relevante para el proyecto, al hacer posible conocer con precisión la distancia de los diferentes puntos del rostro a la cámara.

El sensor de profundidad cuenta con un rango válido que va desde 0.8 hasta 4 metros de distancia, y un rango de reconocimiento facial tentativo que va desde 0.9 hasta 2 metros, tal como se muestra en la figura 2-2.

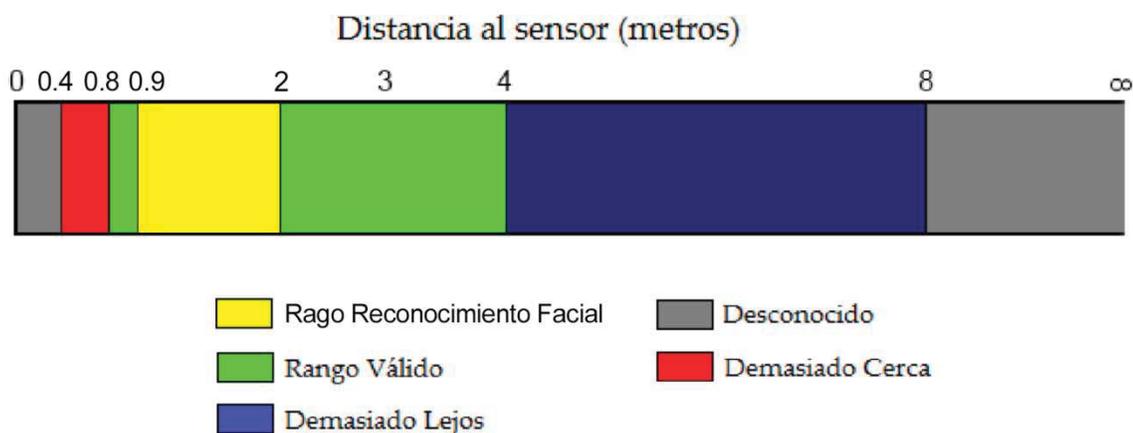


Figura 2-2 Rango Válido del Sensor de Profundidad

La información de profundidad se devuelve en un mapa de píxeles con una frecuencia máxima de 30 imágenes por segundo. Cada píxel está representado por dos bytes (16 bits), cuyo valor representa teóricamente la distancia del objeto al sensor. Si el valor de un píxel del mapa es cero, significa que el sensor no pudo estimar la profundidad para esa región.

La información tridimensional del entorno del sensor es enviada desde el dispositivo Kinect hasta el ordenador a través de un flujo de datos llamado DepthImageStream, que se puede configurar para enviar la información con distintos grados de precisión y tasas de refresco.

Dicha información es almacenada en un buffer denominado DepthImageFrame, que contiene la información de profundidad de un fotograma, por lo que es refrescado cada vez que el sensor captura un nuevo fotograma. Esto implica que si los cálculos a realizar con dicho fotograma necesitan más tiempo para ejecutarse que el tiempo entre fotogramas, se deberá almacenar la información en otro buffer aparte; y probablemente no se podrá analizar la información de los fotogramas producidos durante dichos cálculos en tiempo real.

2.1.4 Usos del dispositivo Kinect

Como ya se mencionó anteriormente el uso del dispositivo Kinect se ha expandido a muchas más áreas de aplicación. Ya se han creado un gran número de proyectos que han ayudado a las personas en sus labores cotidianas. Ya sea en su jornada laboral o extralaboral.

- **Medicina:** Este campo ha mostrado muy importantes avances en la incorporación de la tecnología Kinect. Su uso se ha extendido progresivamente ayudando a mejorar la salud de la personas, su principal avance ha sido en la rehabilitación de personas con discapacidad. Por ejemplo, uno de los más importantes es VirtualRehab. Es un sistema de rehabilitación a distancia que permite realizar las sesiones desde cualquier lugar, mientras que el médico puede consultar en cualquier momento y lugar los avances de los pacientes. Además, el uso de Kinect también ayuda a hacer más llevadera la rehabilitación de los niños, incluidos los afectados por el autismo, ya que los videojuegos pueden resultar bastante divertidos por lo que es más fácil acercarse a ellos.

Otro aporte muy importante se ha visto en los quirófanos españoles que han utilizado este dispositivo para mejorar la forma en que lo cirujanos pueden consultar la información. Esto otorga beneficios, sobre todo a nivel de higiene, ya que el cirujano no debe tener contacto físico con otros elementos, evitando que infecciones y bacterias se puedan transmitir al paciente.

- **Robótica:** Esta es un área en constante crecimiento y ha sabido aprovechar la tecnología Kinect. La Robótica puede beneficiar las demás áreas, un ejemplo muy importante es la colaboración de los Robot con tecnología Kinect que han ayudado a especialista en desactivar bombas, no poniendo en riesgo la vida del especialista. Además existen otras aplicaciones como un “robot mayordomo” que pueden reconocer objetos y recogerlos. Así mismo, existen robots que pueden ayudar a personas que tengas problemas de discapacidad o de momentánea inmovilidad.
- **Industrias Nucleares:** En este campo la utilización de la tecnología Kinect es muy beneficioso ya que ayuda bastante a proteger la salud de los trabajadores que se deben exponer a niveles importantes de radiación. Si bien este sistema en la actualidad no está incorporado completamente existen prototipos con esta finalidad, tal es el caso de un sistema de monitorización de plantas IDbox, sistema creado por la empresa CIC Consulting Informático.
- **Educación:** En el área de la educación la tecnología Kinect tiene mucho potencial, ya que resulta mucho más atractivos para los niños el uso de aplicaciones interactivas que puedan ayudarlos a aprender. Existen muchos

juegos educativos para aprender de forma más interactiva, un ejemplo es” Sky Spelling” que ayuda a reforzar las habilidades de deletreo de los niños mientras simulan ser un avión o un dragón.

2.2 Tecnologías existentes con reconocimiento facial

Aplicar la biometría facial es algo comúnmente usado en el campo de la seguridad, tanto en la identificación de personas que se han visto inmersas en un crimen, así como en la prevención de falsificación de pasaportes, búsqueda de niños perdidos, entre muchas otras.

A continuación se mostraran algunos ejemplos de sistemas que utilizan el reconocimiento facial

- ❖ El sistema de reconocimiento facial de Facebook, llamado DeepFace, desarrollado con la intención de potenciar el etiquetado de fotos.
- ❖ SmartGate es un sistema que pretende sustituir el control de aduanas humano situado en los aeropuertos que se encarga de verificar la identidad de un individuo con su pasaporte
- ❖ Face Unlock desarrollado por la empresa pittpatt perteneciente a google permite a los usuarios de móviles con android cambiar el desbloqueo tradicional por desbloqueo mediante reconocimiento de la cara.
- ❖ Next generation identification programme es un proyecto lanzado por el fbi en 2010 que pretende instaurar un sistema de reconocimiento facial en cámaras de videovigilancia a lo largo de todo estados unidos
- ❖ NameTag desarrollado por la empresa FacialNetwork, un servicio de reconocimiento visual capaz de tomar la fotografía de una persona desconocida, analizar su rostro y buscarlo en cualquier red social en línea.
- ❖ FaceCloud desarrollada por la compañía tecnológica Polar rose, es una solución de reconocimiento facial que se ejecuta en el servidor y permite adicionar el reconocimiento facial a casi todos los servicios controlados a través del servidor y estaba dirigida a los proveedores de servicios web, redes sociales y otras compañías o servicios que conllevan un fuerte uso de fotos digitales.

2.3 Pasos generales en un reconocimiento facial

Para reconocer un rostro se deben realizar los siguientes pasos generales

- Adquisición de la imagen
- Pre-procesamiento de la imagen
- Localización de la cara
- Extracción de características
- Reconocimiento facial

2.3.1 Adquisición de la imagen

Consiste en la captura de una imagen desde un dispositivo, en este proyecto ese dispositivo será la Kinect.

2.3.2 Pre-procesamiento de la imagen

En esta etapa se trata de restaurar y normalizar las imágenes de entrada, a fin de facilitar los procesamientos posteriores y mejorar la calidad de la imagen. Dentro de los pre-procesamientos posibles se pueden incluir mejoras del contraste, ecualizaciones, filtraje para la eliminación de ruido, suavizados, etc.

2.3.3 Localización de la cara

Esta etapa, también llamada etapa de segmentación, localiza la ubicación de la cara de la persona dentro de la imagen capturada, esto se hace con el objetivo de que en la etapa siguiente sólo se incluyan píxeles que pertenezcan a la cara del sujeto. El resultado de esta etapa es una ROI (región de interés) que contenga la cara de la persona.

Para realizar la localización existen muchos métodos propuestos en la literatura, de los cuales el más ampliamente usado es el método de Viola-Jones, el cual consiste en varios detectores en cascada, en cada nivel se analiza una ventana y se decide si la ventana califica o no para pasar al siguiente nivel, si la ventana supera todos los niveles, entonces se determina que la ventana posee una cara, también hay otros métodos que se basan en encontrar el patrón de color de la piel, o en encontrar los ojos y la boca en la imagen, etc [12].



Figura 2-3 Localización y segmentación con Viola-Jones [12]

2.3.4 Extracción de las características faciales

En esta etapa se trata de reducir la cantidad de información que representa a cada uno de los patrones, obteniendo de esta forma, un vector de características que represente de la mejor manera posible al patrón original. Además, cabe señalar que la dimensionalidad del vector de características debe de ser menor que la del patrón original, representando una codificación óptima de la entrada, perdiendo la información que no sea muy importante.

2.3.5 Reconocimiento Facial

En esta etapa, las características extraídas en el paso anterior son llevados a un espacio distinto a través de algún método de reconocimiento facial (vea 4.4), como Análisis de Componentes Principales (PCA, por su sigla en Inglés), Análisis de Correlación, u cualquier otro análisis estadístico de los datos ya extraídos. En otras palabras, consiste en transformar las características faciales (datos) en información que sea representativa de cada persona y que permita su reconocimiento, es decir obtener una salida del sistema como respuesta a un conjunto de estímulos dados como entrada (patrones).

2.4 Técnicas de reconocimiento facial

De todos los rasgos anatómicos, el rostro es el elemento que con más frecuencia utilizamos los seres humanos para identificar a otro individuo y para ello, el cerebro comienza por establecer los aspectos físicos de una cara, a continuación determina si estas facciones son conocidas o no y, por último, procede a otorgar un nombre a lo que ve. Este proceso tan aparentemente sencillo, puede resultar muy difícil para una máquina.

- **Técnicas holísticas.** Estas también se denominan métodos basados en la apariencia o métodos basados en imágenes. Utilizan características de una imagen tales que la describen globalmente, Utilizando la imagen facial completa como patrón, incluido el fondo (aunque es deseable que este ocupe la menor superficie posible), y por tanto se evita la etapa de segmentación. Para mayor detalle de las principales técnicas holísticas (ver Anexo 9.2).
- **Técnicas basadas en rasgos locales.** Son las que utilizan características que describen la cara parcialmente, como ejemplo, un sistema de este tipo, podría consistir en extraer rasgos locales, por ejemplo, ojos, boca, nariz, etc. O bien líneas o puntos, a partir de los cuáles se calculan los valores de las características. Estas pueden consistir en distancias, áreas, ángulos, etc., y sirven para representar a la cara con vistas a la etapa de reconocimiento. Para mayor detalle de las técnicas basadas en rasgos locales (ver Anexo 9.3).

3 Requerimientos

3.1 Requerimientos Funcionales

- [RF01] Permitir registrar usuarios en la base de datos con el dispositivo Kinect.
- [RF02] Reconocer a los usuarios, a pesar de que cuente con accesorio u otros elementos que cambien levemente la apariencia física.
- [RF03] Mostrar una lista con todos los usuarios que estén registrados en la base de datos.
- [RF04] Permitir eliminar usuarios del sistema.
- [RF05] Permitir la modificación de los datos del usuario

3.2 Requerimientos No Funcionales

- El tiempo que demora detección de rostros de usuario debe ser reducido.
- El tiempo de respuesta de la identificación debe ser reducido.
- El nivel de fallos debe ser llevado al mínimo.
- La interfaz debe ser sencilla y minimalista, que ayude durante la investigación.

4 Estudio de Factibilidad

4.1 Factibilidad técnica

Para el desarrollo de este proyecto es necesario cumplir con ciertos aspectos técnicos, tales como:

4.1.1 A nivel de hardware

- Un computador con las siguientes características:
 - Memoria RAM de 2 Gb o superior
 - Procesadores dual core 2.6 Ghz o superior
 - Tarjeta de video que soporte Direct X 9 o superior
- Un dispositivo Kinect

4.1.2 A nivel de software

- Un Sistema operativo Windows 7 o 8/8.1
- Microsoft Visual Studio
- SDK 1.5 para Kinect
- Kinect for Windows Developer Toolkit

4.1.3 A nivel personal

- Conocimiento sobre el lenguaje de programación C#
- Conocimiento en programación orientada a objetos
- Disponibilidad horaria
- Base datos a definir

4.2 Factibilidad Económica

Para que el desarrollo de esta aplicación sea exitoso, se deben tener en consideración los costos por el hardware, software y el personal necesario.

- Costo de hardware: El costo principal será el dispositivo Kinect que será facilitado por el profesor guía. Por lo tanto no se tendrán gastos con respecto a hardware, ya que, se posee lo necesarios para el desarrollo de la aplicación.
- Costo de software: No se Tendrán gastos respecto a software, ya que las herramientas necesarias para el desarrollo de la aplicación son obtenibles de forma gratuita (open source).
- Costo de personal: No se poseen costos asociados.

4.3 Factibilidad Operativa

Visto desde la operatividad, el sistema será semiautónomo, en donde se necesitara personal sólo para el registro o en caso de alguna eventualidad en la detección.

El personal sólo necesita un computador con las características anteriormente mencionadas. Ya que la interfaz será bastante sencilla de utilizar y sin una gran cantidad de funcionalidades asociadas.

4.4 Factibilidad Legal

Este proyecto no cuenta con ningún impedimento legal, ya que el software propuesto es de uso libre. La ley 19.223(Ley de delitos informáticos) no opone ningún impedimento para el desarrollo de este sistema. Además todo usuario que se identifique lo hará por voluntad propia y debe estar registrado en el sistema bajo su consentimiento.

5 Diseño de la Solución

Para la implementación de este proyecto se realizara una aplicación en .NET utilizando el Kit de desarrollo que provee Windows, que permitirá autenticar personas en tiempo real mediante el reconocimiento facial, utilizando la tecnología que provee el dispositivo Kinect. De esta forma se podrá hacer un posterior análisis de los resultados obtenidos, esperando que se cumpla el objetivo de superar la tasa del 95% esperada, incluso llegar a la tasa de los humanos que corresponde al 97.5% de acierto.

El proyecto se implementara mediante la herramienta Visual Studio 2010 que proporciona ayudas al programador, como un diseñador visual de interfaces con el que prácticamente no es necesario programar en lenguaje XAML ya que mediante un sistema Drag and Drop se construye la interfaz de interacción produciendo el código necesario para la ejecución de la aplicación.

La interacción entre el hardware y software con la aplicación se muestra en la figura 5-1.

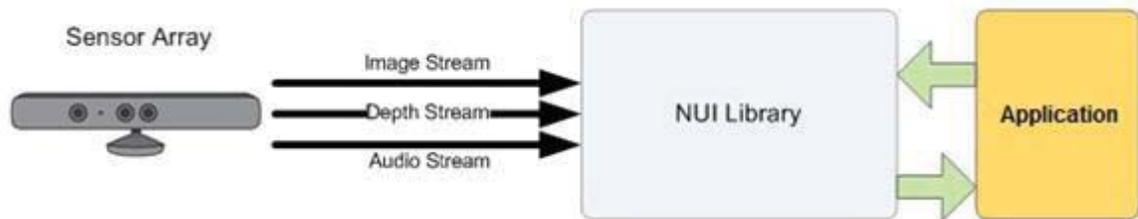


Figura 5-1 Interacción con la aplicación [11]

5.1 Arquitectura de la solución

El desarrollo de esta solución se realizara mediante un arquitectura de tres capas que está compuesta por la capa de presentación, negocio y de datos. A continuación se explicara cada una de las funciones del capaz del sistema [3]:

- **Capa Presentación:** Es la parte visible del sistema para el usuario, en donde se presenta la información y en donde se solicitan las entradas y se muestran las salidas. En este caso se mostrara al administrador una interfaz que permitirá enviar y solicitar información con el sistema. Además debe ayudar al correcto posicionamiento para poder realizar la captura con el dispositivo Kinect
- **Capa de Negocio:** Esta capa contiene toda la lógica del sistema y es en donde se establecen todas las reglas que deben cumplirse. Esta capa interactúa con la capa en presentación, recibiendo las solicitudes del usuario y enviando las respuestas tras el proceso. Además, se comunica con la capa de datos para recuperar o almacenar datos de la base de datos. En este caso la capa de negocio tendrá como principal función, ejecutar los algoritmos necesarios para la correcta identificación de la persona utilizando la librería NUI para la interactuar con el dispositivo kinect.
- **Capa de Datos:** Es en donde se reciben y se accede a los datos. Por lo general está formado por una o más gestores de bases de datos que se encargan del

almacenamiento de datos. Esta capa interactúa con la capa de negocios recibiendo y recuperando datos. En este caso la base de datos almacenara los datos de los usuarios, así como los parámetros que se necesitaran para su posterior identificación

A continuación se muestra la interacción entre las 3 capas de la aplicación

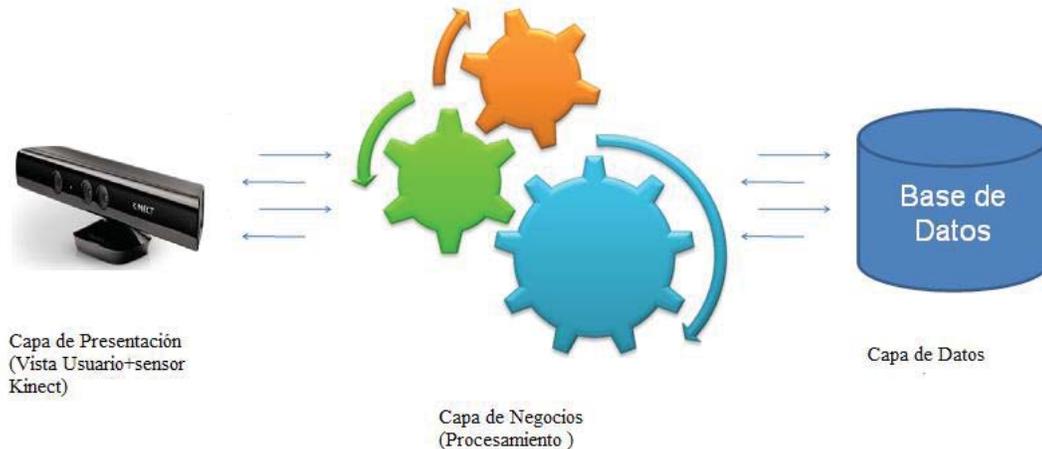


Figura 5-2 Estructura de la solución

5.2 C Sharp (C#)

El lenguaje de programación C Sharp, también llamado C# es un lenguaje de programación orientado a objetos desarrollado por Microsoft e incluido en la plataforma .NET. Emplea una sintaxis derivada de C y C++, aunque C Sharp fue diseñado como una mejora de ambos lenguajes.

Este lenguaje ha supuesto uno de los pilares fundamentales en la elaboración del proyecto, ya que su potencia visual y su gran sinergia con el SDK de Kinect lo convierten en el lenguaje más adecuado para la elaboración de las herramientas de adquisición de la información y de análisis de resultados.

El lenguaje C Sharp, junto con el entorno de desarrollo Microsoft Visual Studio, permite el desarrollo de aplicaciones visualmente atractivas de una forma rápida, sencilla e intuitiva.

5.3 Interfaz de Seguimiento Facial de Kinect (Face Tracking)

Como ya lo habíamos mencionado anteriormente, el kit de desarrollo de software del dispositivo Kinect (SDK), otorga una API de Face tracking, que permite identificar los 121 puntos del rostro, siendo capaz de poder seguirlos mientras el rostro se encuentre dentro del

campo visual del dispositivo. Cada punto posee ciertas características, siendo los factores de posición X, Y, Z lo más relevante. Este conjunto de puntos, crean una maya que es única para cada persona

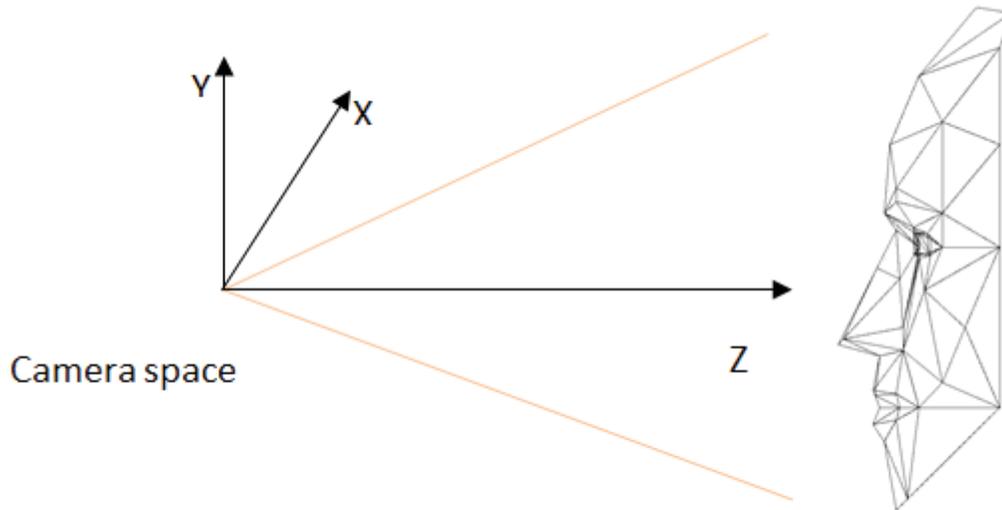


Figura 5-3 FaceTracking

Teóricamente la Kinect debería ser capaz de establecer los 121 puntos y poder mantener el mapeo de los puntos aún en movimiento. A continuación se realiza una comparación de una captura teórica con una captura real.

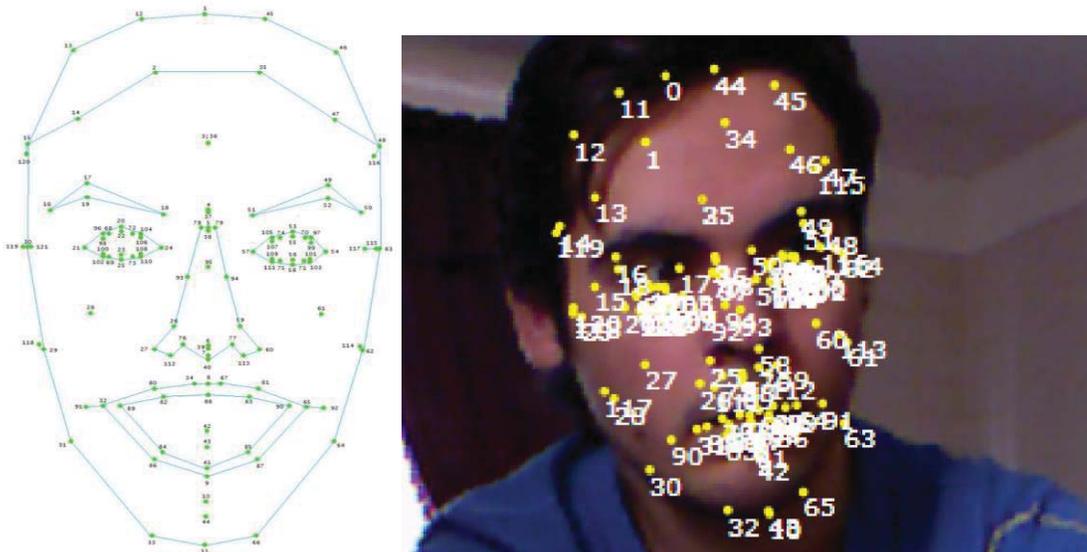


Figura 5-4 Comparación FaceTracking

5.3.1 Descripción de la API

La API de Seguimiento Facial de Kinect fue diseñada para funcionar en tiempo real a partir de los datos que es capaz de recoger esta cámara. Para poder funcionar correctamente tiene una serie de interfaces, que son los encargados de extraer, actualizar y procesar la información obtenida por el dispositivo Kinect. A continuación se muestran las principales Interfaces de la API de Kinect.

Interfaz	Descripción
IFTFaceTracking	Interfaz principal de seguimiento facial
IFTResult	Resultado de un proceso de seguimiento
IFTImage	Una interfaz de apoyo que agrupa varios buffers de imágenes
IFTModel	Un modelo 3D ajustado al rostro seguido

Tabla 2 Interfaces de la API

Para más detalle de los métodos nombrados ver Anexo 9.4.

5.4 ¿Por qué Kinect?

Existen muchos dispositivos y cámaras que se pueden usar para lograr un reconocimiento facial. Pero el dispositivo Kinect posee ciertas ventajas que la hacen sobresalir ante otros dispositivos a la hora de desarrollar una aplicación para el reconocimiento facial.

Una ventaja que provee este dispositivo es la cámara de profundidad que envía y recibe infrarrojo. Lo que permite obtener capturas 3D y de esta manera tener una mejor percepción de la biometría del rostro, además de esto, los factores de eliminación de luz no son tan determinantes como con las cámaras tradicionales.

Otra ventaja es que Microsoft provee un Kit de desarrollo de software (SDK), ofreciendo una serie de librerías, para facilitar el trabajo de los programadores, ya que contiene archivos de cabecera, biblioteca, muestras, documentación, que utilizan las API necesarias para desarrollar aplicaciones para Windows.

La ventaja más considerable que entrega la Kinect es el Face Tracking que provee el SDK, que a través de la cámara de profundidad es capaz de detectar 121 puntos característicos del rostro, que pueden usarse como base para empezar a desarrollar un sistema de reconocimiento facial.

Además el dispositivo Kinect es mucho más asequible económicamente que otros dispositivos y cámaras de mayor valor en el mercado. Si bien, se puede conseguir desarrollar un sistema de reconocimiento facial con cámaras de menor costo, estas no ofrecen las ventajas que ofrece este dispositivo y sus resultados no son lo suficientemente buenos.

5.5 Técnica Utilizada

Al poder contar los puntos característicos del rostro que es capaz de detectar y seguir el dispositivo Kinect a través del Face Tracking, se utiliza una técnica basada en rasgos locales, en donde se podrán diferenciar ciertas características a través de las cualidades del grafo obtenido.

Este proyecto apunta a crear un algoritmo que sea capaz de obtener las principales características faciales, que se mantengan fijas o con muy poca variación en el rostro y que en conjunto sean capaces de diferenciar a cada una de las personas. Estas características son llamadas “características de forma única” (Shape Unit), las cuales se consideraron como base para la determinación de los puntos más representativos y fiables encontrados, a lo largo de pruebas realizadas.

Características de forma única

- Altura de la cabeza
- Posición vertical de las cejas
- Posición vertical de los ojos
- Altura de los ojos

- Anchura de los ojos
- Separación de los ojos
- Posición vertical de la nariz
- Posición vertical de la boca
- Anchura de la boca
- Distancia vertical entre ojos
- Anchura de la barbilla

La técnica utilizada consiste en ubicar a la persona a una distancia de 1,65 mt y al hacer click en “iniciar reconocimiento” comienza la captura de 36 frames a lo largo de 3 segundos, los cuales se guardan en una matriz de 3 dimensiones (la dimensión frame, la dimensión puntos que va de 0 a 120 y la dimensión eje que podía ser x,y,z). Luego de almacenar los frame en la matriz, se calcula el frame promedio.

Luego para eliminar la dispersión de los datos se hace una comparación de los frames guardados con el frame promedio, eliminándose los frames que se alejan de mayor manera al promedio, recalculando el frame promedio con los frames restantes.

Para finalizar se aplica una técnica de Promedio Usando Áreas y Distancias (ver Resultados de identificación obtenidos 9.2), la cual calcula distancias y áreas del frame promedio con menor dispersión mencionado anteriormente; luego se calcula la diferencia entre el patrón entrante y el patrón almacenado y si la diferencia es menor al umbral de variación fijado para cada patrón, se aumenta el contador coincidencias. Luego si la sumatoria de coincidencias es mayor al umbral de coincidencias definido, significaría que la persona es quien dice ser.

Es importante establecer un umbral de variación apropiado, ya que al ser muy grande, es posible que se llegue a una tasa muy alta de falsos positivos, y si por el contrario, se decide establecer un umbral considerablemente bajo, la comparación sería muy inflexible a la hora de realizar la captura, impidiendo que se identifique la persona si su cara es levemente diferente a la cara almacenada en la base de datos

El umbral de variación de cada patrón se calculó a partir de pruebas con usuarios. Calculando el promedio de las desviaciones estándar de cada patrón utilizados. Esto se puede representar como:

$$U = \frac{\sum_{i=1}^n S_i}{n}$$

Siendo n la cantidad de usuarios y S la desviación estándar.

En este Pseudocódigo se explica de manera general el sistema de reconocimiento facial que se propone.

```
INICIO
Mientras(captura)
{
  SeleccionarUsuario(Usuario);
  Si (deteccionCara(captura) && IniciarReconocimiento())
  {
    ListaFrame []+=getFrame();
    Si (ListaFrame.Size==36) `
    {
      PatronPromedio=CalcularPromedio(ListaFrame);
      PatronPromedio=EliminarDispersion(PatronPromedio);
      RecorrerPatrónPromedio
      {
        ResultadoDiferencia = CompararPatrón(Patrón,PatrónUsuario);
        Si(ResultadoDiferencia<UmbralDeVariación)
          ResultadoCoincidencias++;
      }
      Si (ResultadoCoincidencias > umbral)
        reconocimiento_satisfactorio();
      Sino
        no_identificado();
    }
  }
}
FIN
```

5.6 Casos de Uso

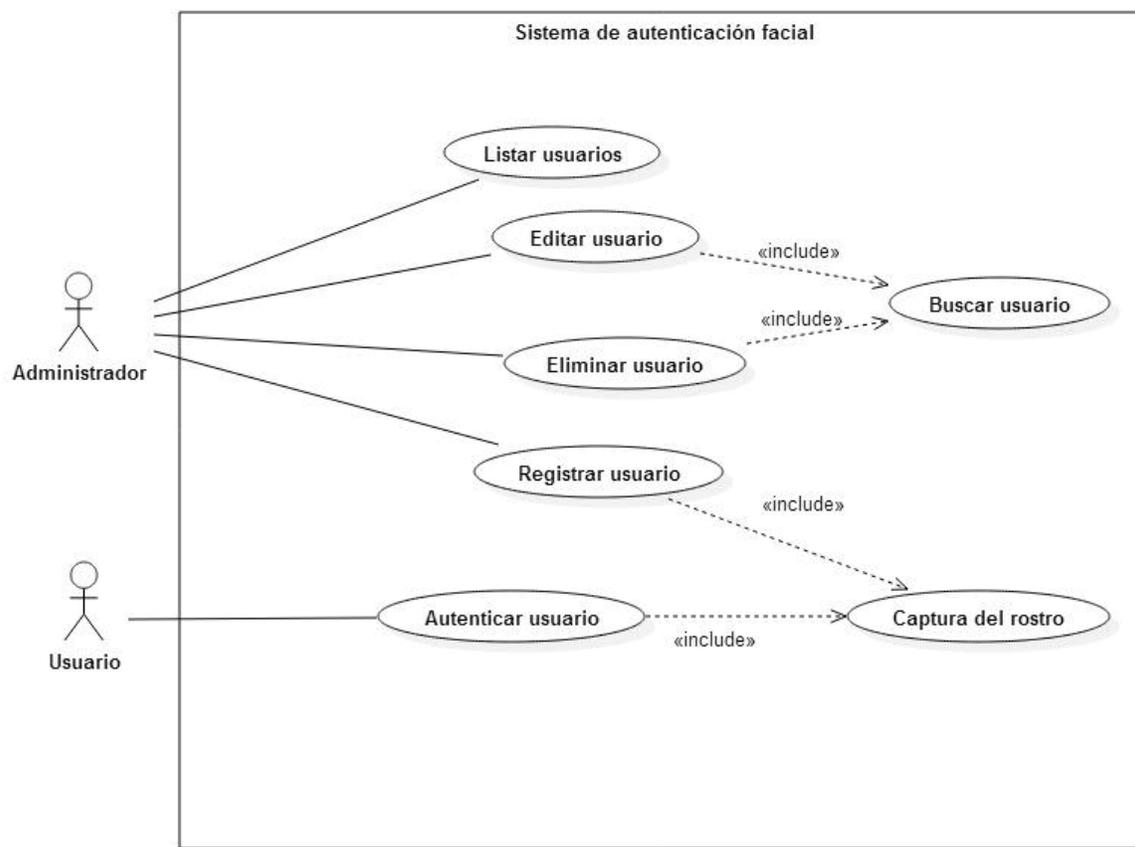


Figura 5-5 Casos de uso

En el diagrama anterior se puede apreciar las principales funcionalidades y las interacciones que existen con los actores. Existen dos tipos de actores, el administrador que tendrá que intervenir ocasionalmente y el usuario normal que será la persona que busca autenticarse.

5.6.1 Autenticar Usuario

Caso de uso	1. Autenticar Usuario
Actor Principal	Usuario
Descripción	Habilitar el acceso al sistema al verificar que la persona identificada está registrada
Precondiciones	Que el rostro del usuario este registrado
Post-condiciones	Usuario ingresa exitosamente.

Escenario principal	<ol style="list-style-type: none"> 1. El usuario ingresa persona a identificar. 2. El sistema detecta un rostro. 3. El sistema le hace una captura al rostro detectado. 4. El sistema verifica si el rostro del usuario pertenece al sistema 5. El sistema lo encuentra, habilitándole el acceso al lugar
Escenario Secundario	<ol style="list-style-type: none"> 3.1- La captura no fue realizada correctamente. <ol style="list-style-type: none"> 3.1.1- El sistema despliega mensaje de aviso. 3.1.2- Volver al paso 2.
Frecuencia	Alta
Referencia	RF02
Observaciones	No posee.

Tabla 3 Caso de uso Autenticar Usuario

5.6.2 Registrar Usuarios

Caso de uso	2. Registrar Usuarios.
Actor Principal	Administrador
Descripción	Permite registrar a un usuario en la base de datos
Precondiciones	El administrador debe haber ingresado a la opción “Registrar usuarios”
Post-condiciones	El administrador registra a una persona en el sistema exitosamente.
Escenario principal	<ol style="list-style-type: none"> 1. El sistema detecta un rostro y coloca un cuadro. 2. El sistema hace una captura del rostro. 3. El administrador digita los datos de la persona 4. El administrador guarda los cambios
Escenario Secundario	<ol style="list-style-type: none"> 2.1- La captura no fue realizada correctamente. <ol style="list-style-type: none"> 2.1.1- El sistema despliega mensaje de aviso. 2.1.2- Volver al paso 1.
Frecuencia	Alta
Referencia	RF01
Observaciones	No posee.

Tabla 4 Caso de uso Registrar Usuarios

5.6.3 Captura del rostro

Caso de uso	3. Captura del rostro
Actor Principal	Usuario
Descripción	Capturar el patrón facial a través del sensor Kinect.
Precondiciones	No posee.
Post-condiciones	Captura del rostro exitosamente.
Escenario principal	<ol style="list-style-type: none"> 1. El sistema detecta un rostro. 2. El sistema le hace una captura al rostro detectado.
Escenario Secundario	<ol style="list-style-type: none"> 2.1- La captura no fue realizada correctamente. <ol style="list-style-type: none"> 2.1.1- El sistema despliega mensaje de aviso. 2.1.2- Volver al paso 1.
Frecuencia	Alta
Referencia	No posee
Observaciones	No posee.

Tabla 5 Caso de uso Capturar rostro

5.6.4 Listar usuarios

Caso de uso	4. Listar usuarios
Actor Principal	Administrador
Descripción	Muestra a los usuarios existentes en la base de datos que tienen acceso al lugar.
Precondiciones	El administrador debe haber ingresado a la opción “Listar usuarios”
Post-condiciones	El Sistema muestra exitosamente a los usuarios.
Escenario principal	<ol style="list-style-type: none"> 1. El sistema busca a los usuarios con acceso a el lugar 2. El sistema muestra los usuarios
Escenario secundario	No posee

Frecuencia	Baja
Referencia	RF03
Observaciones	No posee.

Tabla 6 Caso de uso Listar Usuarios

5.6.5 Editar usuario

Caso de uso	5. Editar usuario
Actor Principal	Administrador
Descripción	Permite editar a un usuario en la base de datos
Precondiciones	El administrador debe estar dentro de la opción “Editar usuario”
Post-condiciones	El administrador edita una persona en el sistema exitosamente.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario selecciona a la persona a editar en el listado. 2. El sistema entrega los datos del usuario a editar. 3. El usuario modifica los datos de la persona 4. El sistema guarda los cambios
Escenario Secundario	<ol style="list-style-type: none"> 4.1 No se pueden guardar los cambios <ol style="list-style-type: none"> 4.1.1- El sistema despliega mensaje de aviso.
Frecuencia	Baja
Referencia	RF05
Observaciones	No posee.

Tabla 7 Caso de uso Editar Usuario

5.6.6 Eliminar usuario

Caso de uso	6. Eliminar usuario
Actor Principal	Administrador
Descripción	Permite eliminar a un usuario en la base de datos
Precondiciones	El administrador debe haber ingresado a la opción “Eliminar usuario”
Post-condiciones	El administrador elimina a una persona en el sistema exitosamente.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario selecciona a la persona a eliminar en el listado. 2. El sistema entrega los datos del usuario a eliminar.

	3. El usuario selecciona la opción eliminar. 4. El sistema guarda los cambios.
Escenario Secundario	4.1 No se pueden guardar los cambios 4.1.1- El sistema despliega mensaje de aviso.
Frecuencia	Baja
Referencia	RF04
Observaciones	No posee.

Tabla 8 Caso de uso Eliminar Usuario

5.6.7 Buscar usuario

Caso de uso	10. Buscar usuario
Actor Principal	Administrador
Descripción	Se filtra dentro de los usuarios a uno en particular.
Precondiciones	
Post-condiciones	El Sistema encuentra a un usuario exitosamente.
Escenario principal	1. El usuario busca a la persona en el sistema. 2. El sistema la encuentra y entrega sus datos.
Escenario secundario	No posee
Frecuencia	Baja
Referencia	No posee
Observaciones	No posee.

Tabla 9 Caso de uso Buscar usuario

5.7 Diagrama de Clases

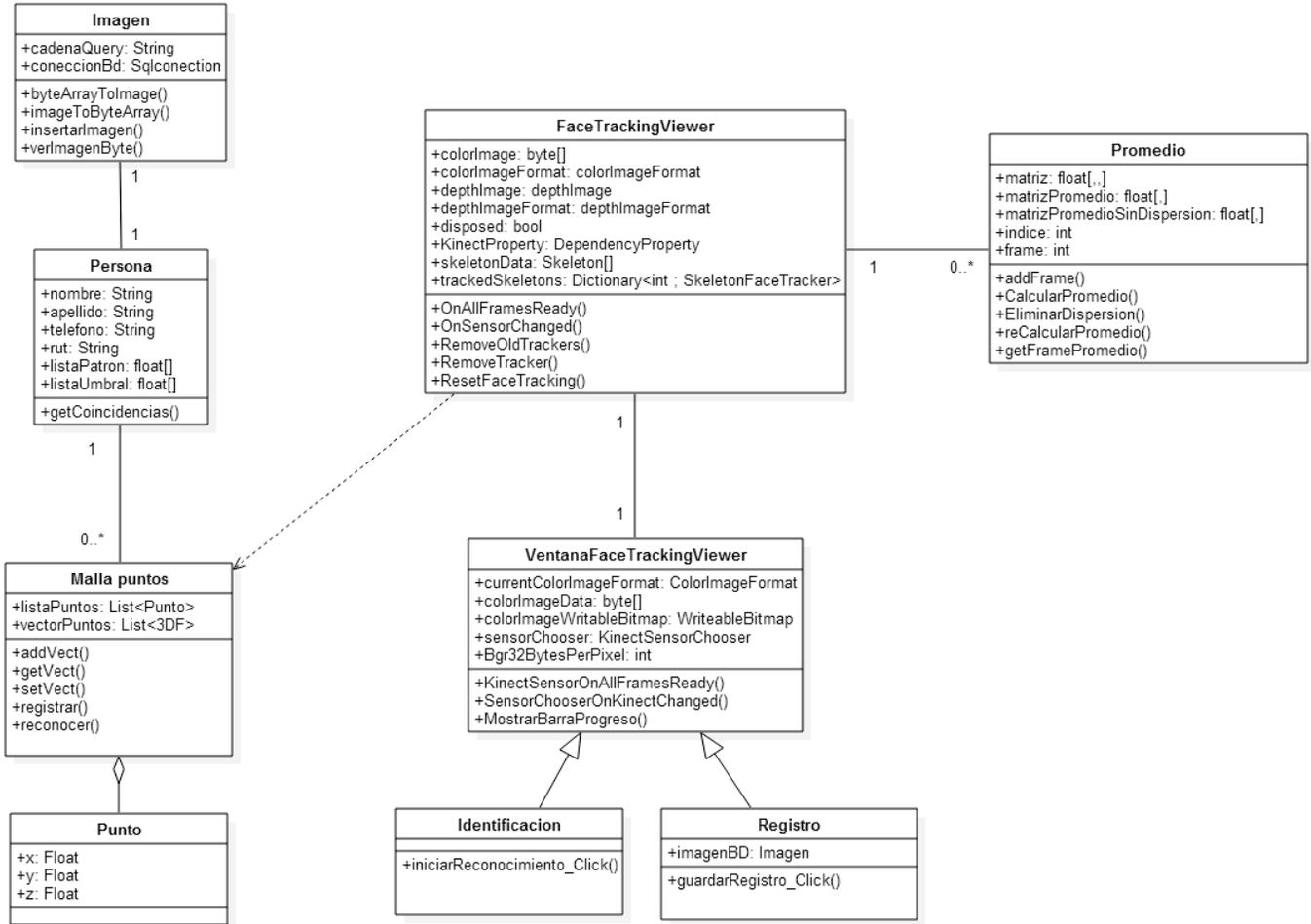


Figura 5-6 Diagrama de Clases

5.8 Diagramas de Secuencia

En un diagrama de secuencia se muestra gráficamente los eventos que influyen de los actores al sistema. Además muestra como el sistema responde a dichos eventos. Son muy útiles para conocer de una forma general como trabajara el sistema frente a determinado evento.

5.8.1 Registrar Usuario

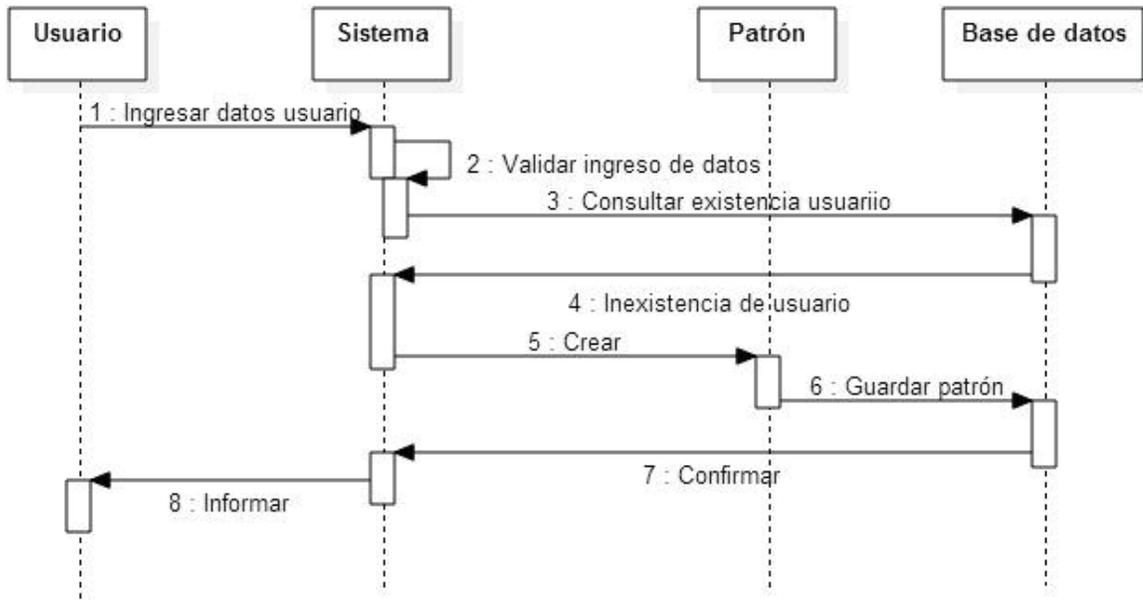


Figura 5-7 Diagrama de secuencia Registro de usuario

En la figura 5-7 se muestra el diagrama de secuencia se muestra gráficamente los eventos que influyen de los actores al sistema en la funcionalidad Registro de usuario.

5.8.2 Autenticar Usuario

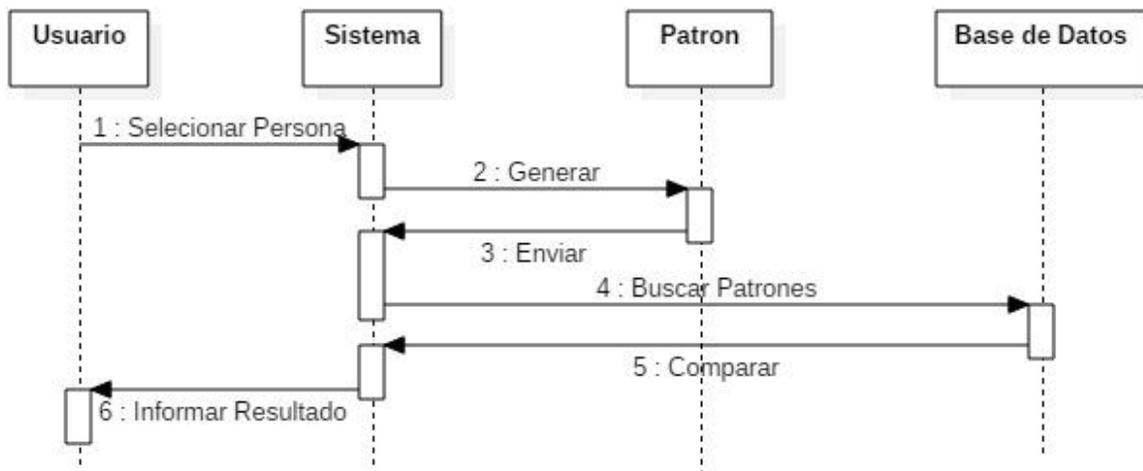


Figura 5-8 Diagrama de secuencia Autenticar usuario

En la figura 5-8 se muestra el diagrama de secuencia se muestra gráficamente los eventos que influyen de los actores al sistema en la funcionalidad Autenticar usuario.

5.8.3 Editar usuario

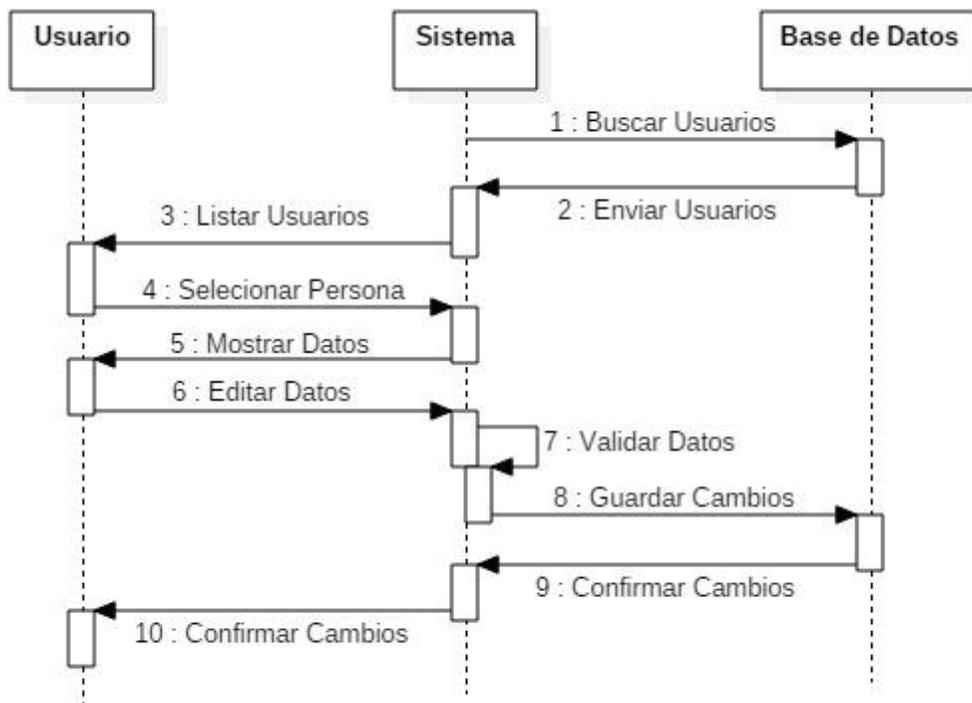


Figura 5-9 Diagrama de secuencia Editar usuario

En la figura 5-9 se muestra el diagrama de secuencia se muestra gráficamente los eventos que influyen de los actores al sistema en la funcionalidad Editar usuario.

5.9 Diagramas de actividad

En un diagrama de actividades se muestra un proceso de negocio o un proceso de software como un flujo de trabajo a través de una serie de acciones. Estas acciones las pueden llevar a cabo personas, componentes de software o equipos. A continuación se mostraran algunos de los diagramas más importantes de este proyecto.

5.9.1 Registro de usuario

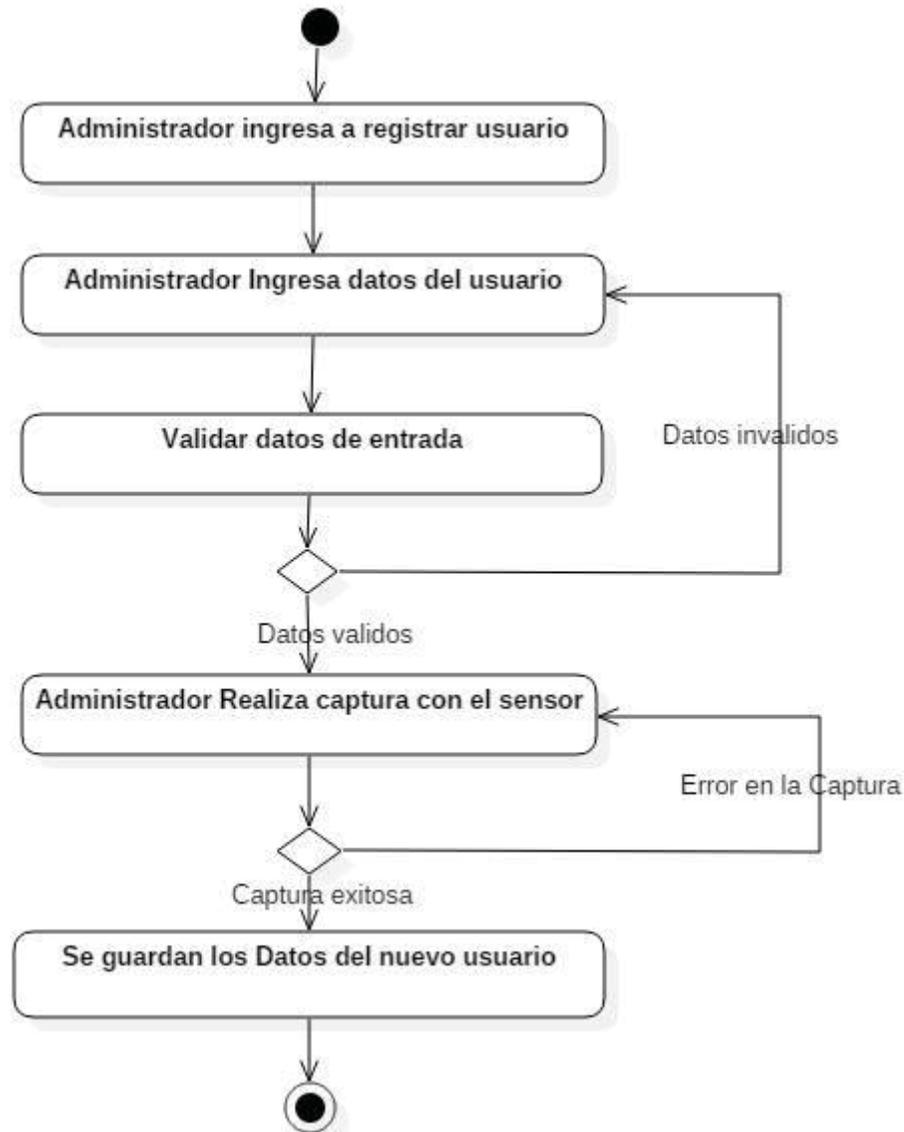


Figura 5-10 Diagrama de actividad Registro de usuario

5.9.2 Autenticar Usuario

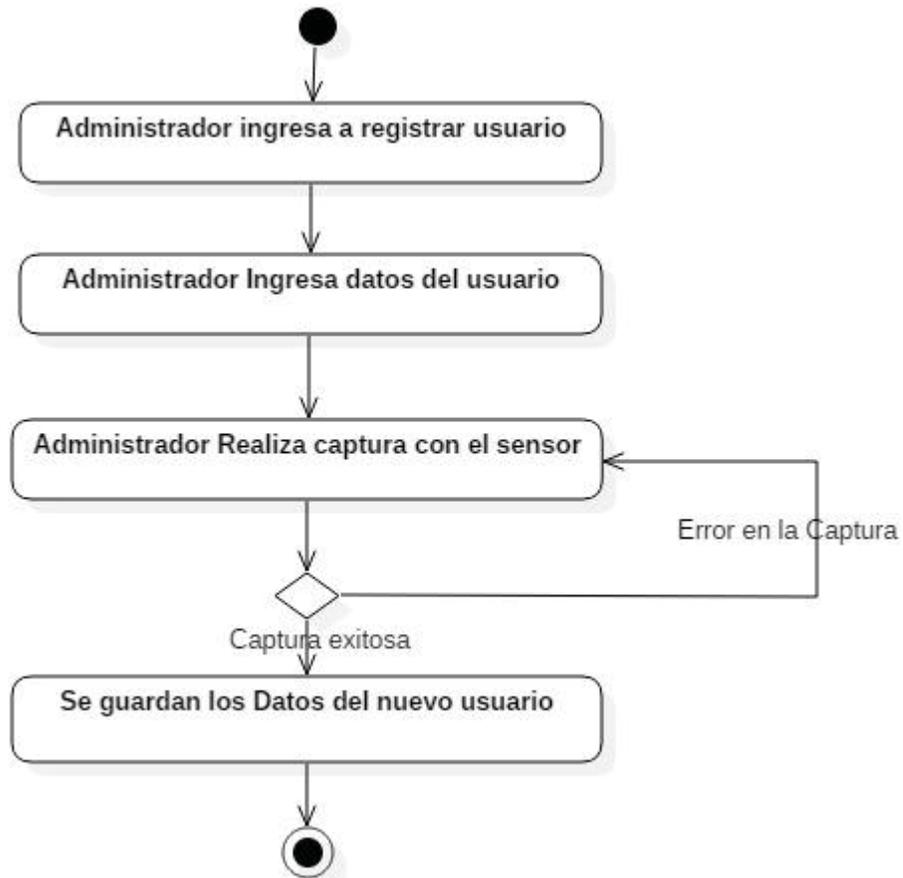


Figura 5-11 Diagrama de actividad Autenticar Usuario

5.10 Modelo Relacional

Usuario	
PK	rut
	nombre
	apellido
	telefono
	patron1
	patron2
	patron3
	patron4
	patron5
	patron6
	patron7
	patron8
	patron9
	patron10
	patron11
	patron12
	patron13
	patron14
	patron15
	Imagen

Figura 5-12 Modelo Relacional

6 Implementación de la solución

6.1 Prototipo Final

6.1.1 Menú

El menú Principal cuenta con tres funcionalidades, iniciar reconocimiento, registrar usuario y listar usuarios, las cuales serán detalladas a continuación.

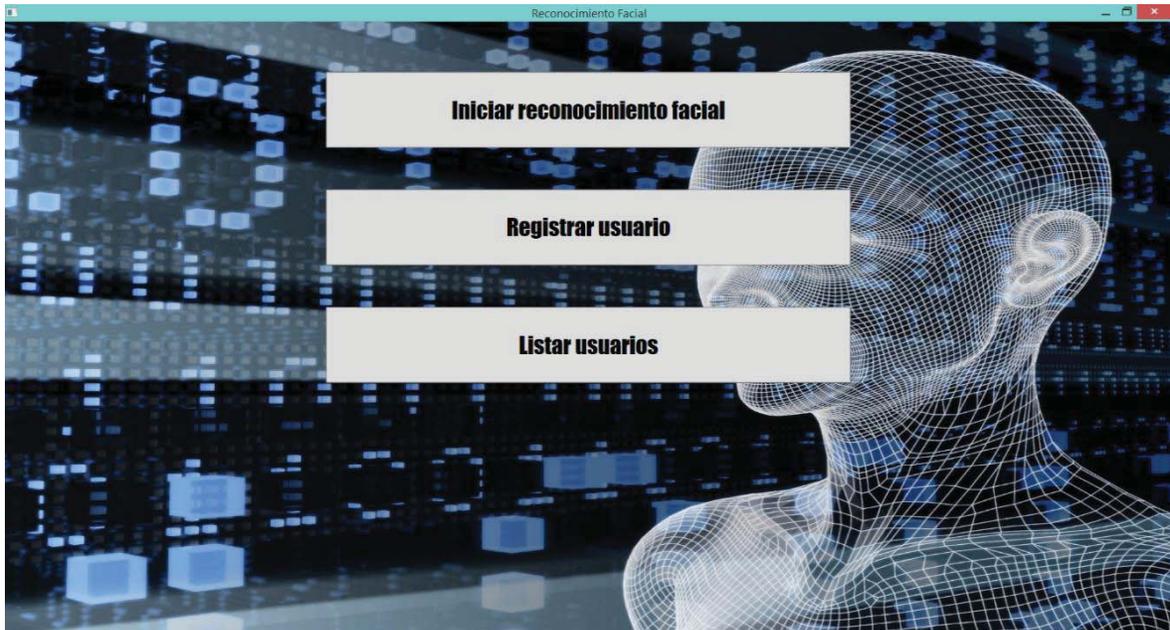


Figura 6-1 Menú

6.1.2 Inicio de reconocimiento facial

Esta funcionalidad se puede reconocer a los usuarios registrados. Hay una lista, en donde se debe seleccionar al usuario a autenticar. El recuadro indica la posición aproximada que debe tener el rostro (ver figura 6-2). El proceso dura 3 segundos, usando una barra de carga para indicar que se está procesando (ver figura 6-3). Finalmente si el reconocimiento fue exitoso el recuadro se torna verde y se muestra un mensaje (ver figura 6-4)

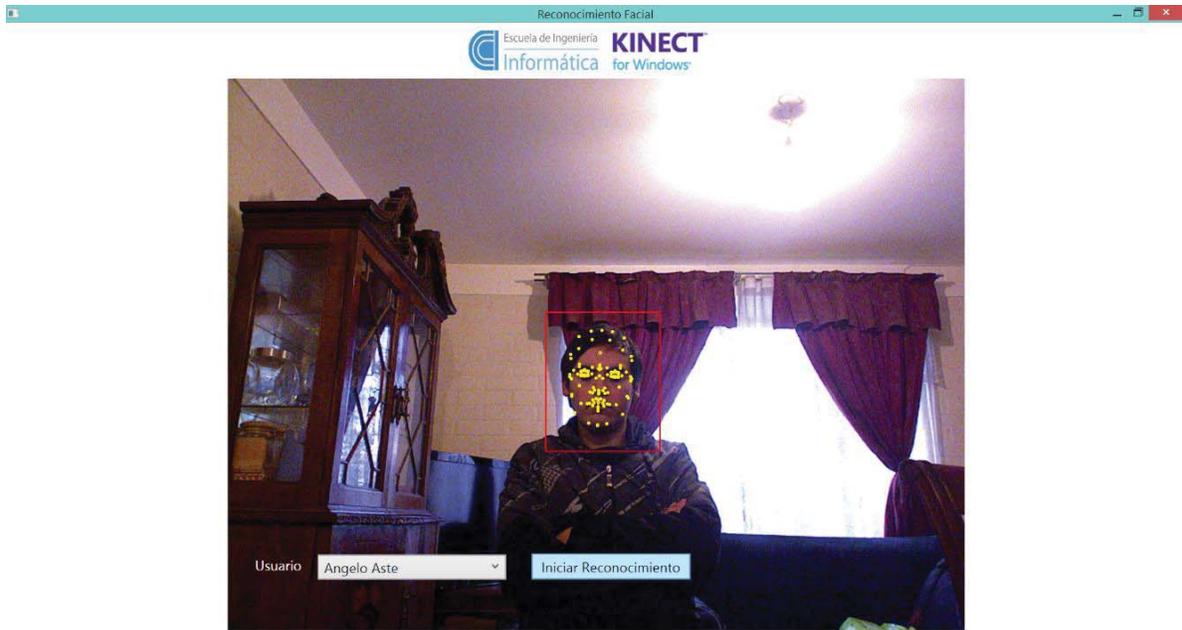


Figura 6-2 Inicio de reconocimiento

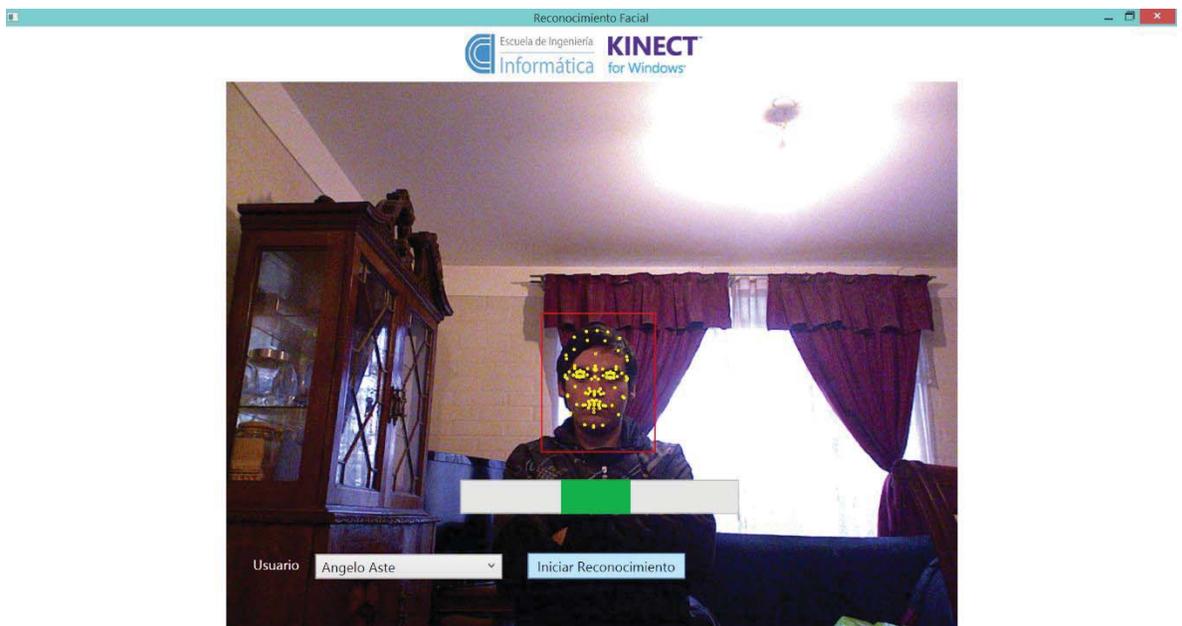


Figura 6-3 Proceso de captura

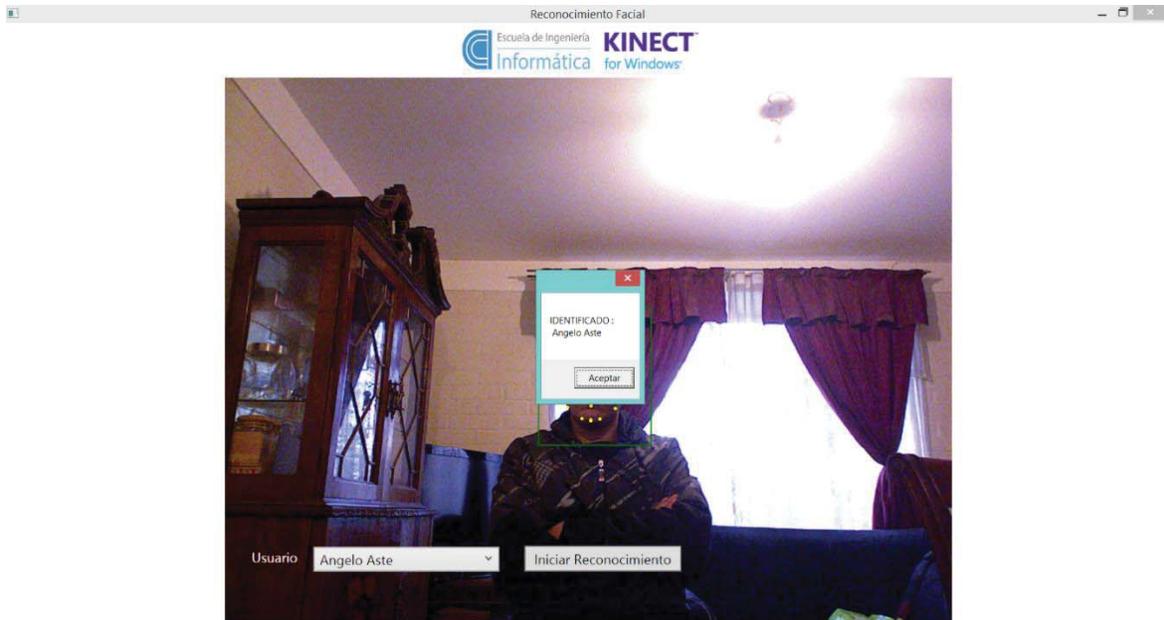


Figura 6-4 Reconocimiento exitoso

6.1.3 Registrar usuario

En esta funcionalidad se permite ingresar nuevos usuarios al sistema. Para esto se debe llenar un formulario (ver figura 6-5). Luego de llenar los campos, se debe realizar una captura del rostro para ser registrado (ver figura 6-6). Al lograr registrar el usuario, el recuadro se torna verde (ver figura 6-7).

The image shows a software window titled "Reconocimiento Facial - Registro de usuario". The background features a wireframe model of a human head. On the left side, there are four labels: "Rut", "Nombre", "Apellido", and "Telefono". To the right of each label is a white rectangular input field. Below these fields is a button labeled "Registrar".

Figura 6-5 Formulario de registro

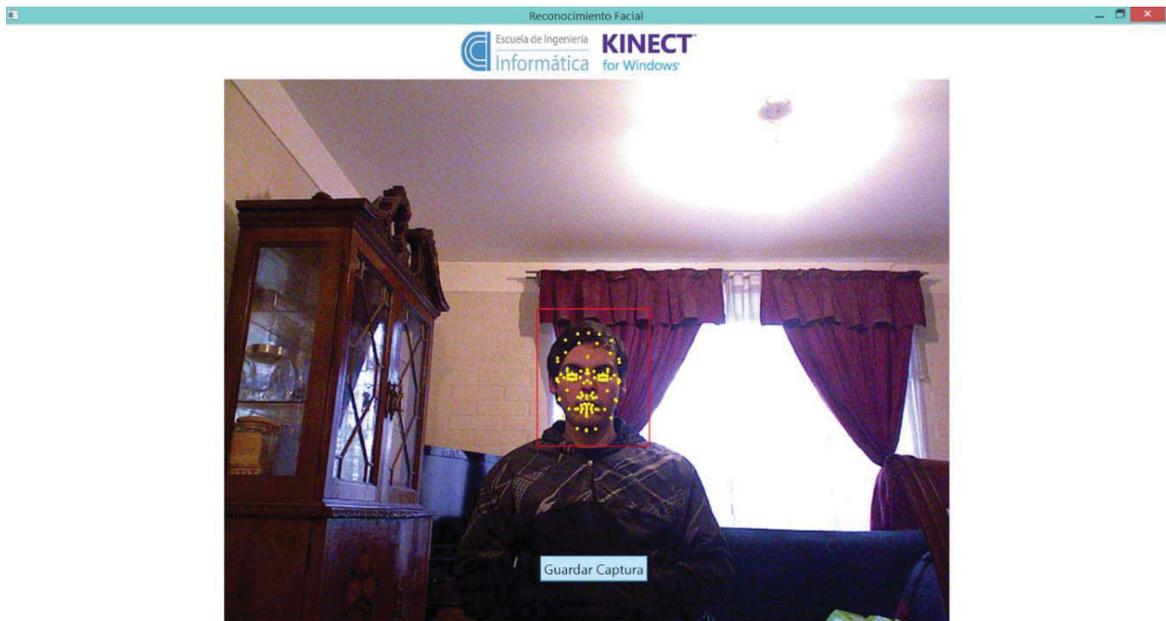


Figura 6-6 Captura del rostro

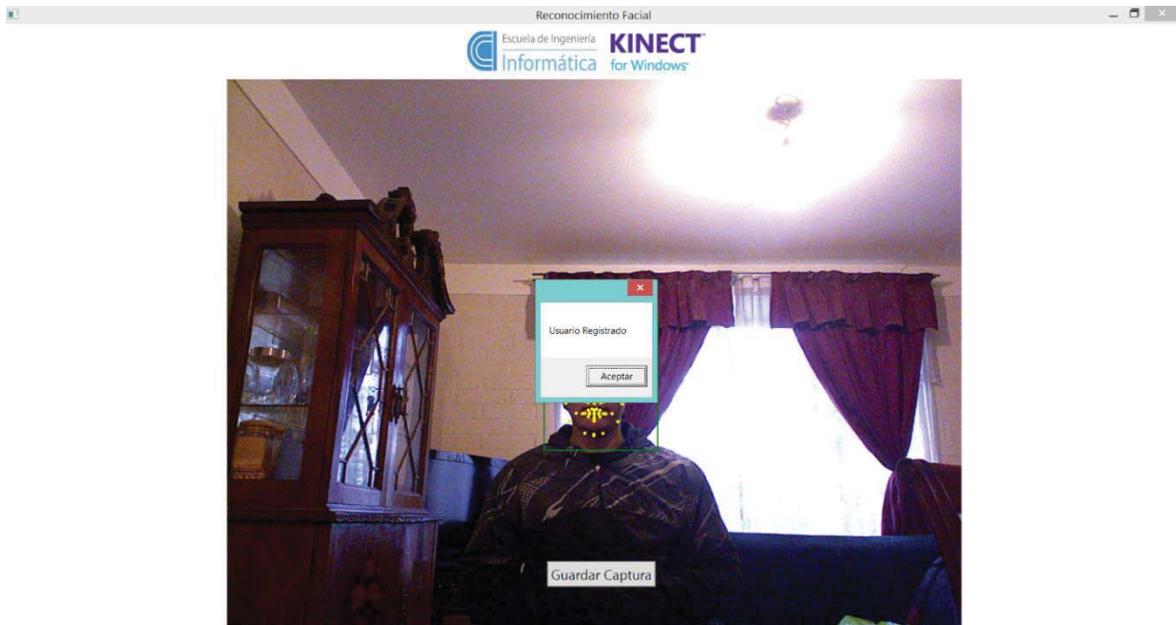


Figura 6-7 Captura exitosa

6.1.4 Listar usuarios

En esta funcionalidad se permite visualizar todos los usuarios registrados. Pudiendo observar, su foto obtenida por la cámara RGB al momento del registro y sus datos personales.

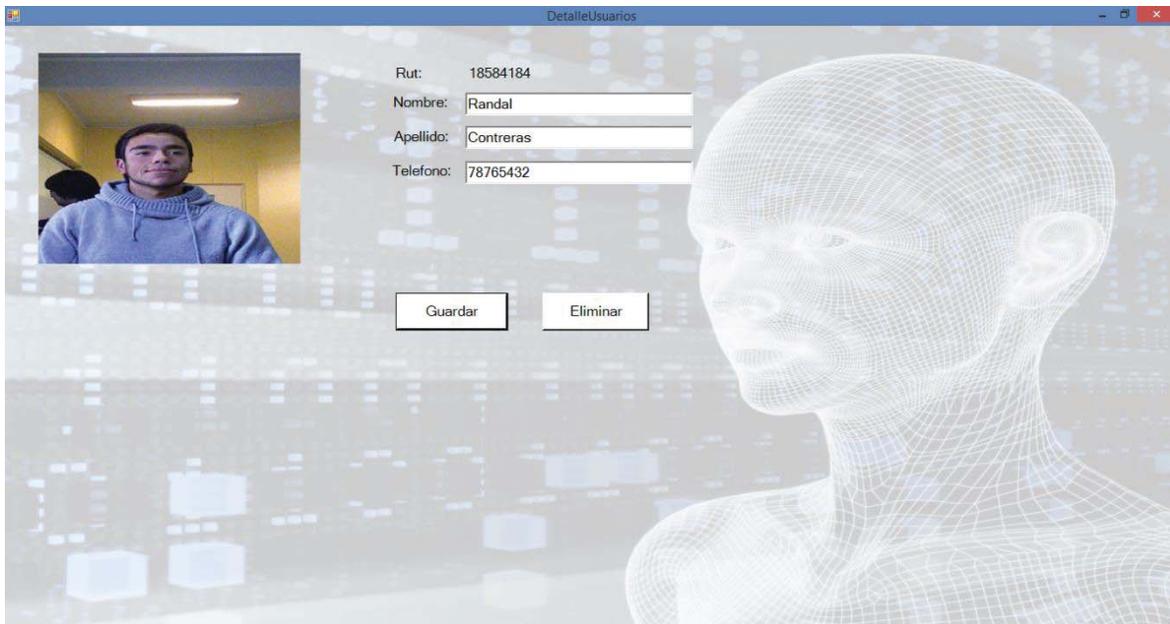


Imagen	Rut	Nombre	Apellido	Telefono
	17964944	Consuelo	Figueroa	75604287
	17719506	Cristofer	Montiel	95202227
	18104266	Diego	Ruiz	99765432

Figura 6-8 Listar usuarios

6.1.5 Detalle Usuario

Al seleccionar un usuario, es posible editar los datos de registro, en caso que tengan que ser modificados. Además es posible eliminar el usuario del Sistema en esta misma pantalla.



Rut: 18584184
Nombre: Randal
Apellido: Contreras
Telefono: 78765432

Figura 6-9 Detalle usuario

6.2 Problemas y limitaciones observadas

Los resultados obtenidos para este avance de proyecto no fueron muy satisfactorios, ya que el sensor Kinect una vez que logra hacer un seguimiento del rostro puede presentar fallas en los puntos estimados, implicando que estos puntos tengan un margen de error y que la fiabilidad de los datos no sea del todo aceptable a la hora de considerarlos para un reconocimiento facial. Esta poca fiabilidad de los datos se debe a las limitaciones presentes en el dispositivo, de las que se hablarán a continuación.

Estas limitaciones vienen condicionadas tanto por factores internos, debidos a la arquitectura del dispositivo; como externos, debidos a la naturaleza de la escena. Dentro de factores internos se tienen las siguientes limitaciones: los puntos de luz no cubren de forma continua la superficie del rostro, lo que conlleva a que algunos píxeles de la imagen de profundidad tengan que ser interpolados generando que el valor de profundidad de un píxel determinado tenga asociado un margen de error. Este margen es mayor cuanto más alejado está el objeto, puesto que, para una misma superficie, los puntos de luz están más separados. A mayores distancias, los valores de profundidad devueltos para objetos cercanos entre sí tienden a ser muy similares. Sin embargo, si el objeto está a demasiada distancia del sensor, no se calcula ninguna distancia para ese punto. Esto ocurre así, porque la potencia de luz del haz de infrarrojos se atenúa en el trayecto recorrido, haciendo que sea imperceptible para el sensor de infrarrojos.

Para lograr determinar cuáles eran los puntos más fiables y representativos del rostro se hicieron pruebas en las que se evaluaba un real seguimiento del rostro a través de movimientos y gestos faciales. (Ver figura 6-10)

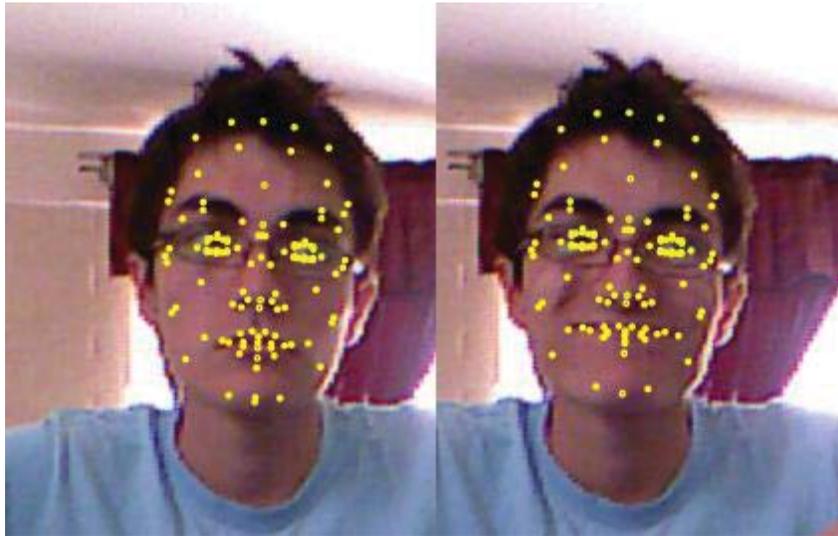


Figura 6-10 Seguimiento del rostro

Se consideraron las siguientes distancias entre puntos 2-49, 13-65, 50-48, 16-53 (ver figura 6-11) (perteneciente a los puntos más fiables determinados en una fase anterior) las cuales se usaron para determinar la mejor distancia del rostro al sensor a la hora de hacer una identificación de un rostro.

Haciendo pruebas aplicadas a las siguientes distancias al dispositivo kinect:

1.2 metros, 1.5 metros, 1.65 metros, 1.8 metros

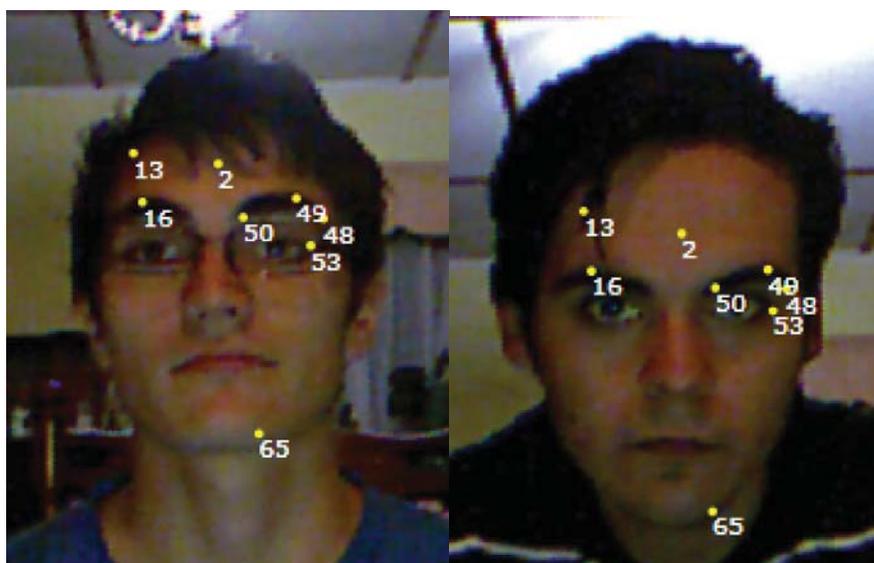


Figura 6-11 Puntos considerados

A continuación se mostraran las tablas con la información de que tanto varían los datos en relación a la distancia del rostro al sensor, con el objetivo de lograr saber que distancia era la más adecuada para realizar una identificación facial.

Distancias del rostro al sensor	Desviación estándar Alumno 1	Desviación estándar Alumno 2	Desviación estándar Promedio de alumnos	Tiempo para reconocer 121 puntos del rostro	Desviación estándar *tiempo
1,2 mts	0,00412566	0,00340354	0,0037646	14,13 segundos	0,0531938
1,5 mts	0,0022629	0,0050192	0,00364105	4,658 segundos	0,01696001
1,65 mts	0,0023455	0,00595412	0,00414981	1,216 segundos	0,00504617
1,8 mts	0,00368472	0,00907784	0,00638128	1,148 segundos	0,00732571

Tabla 10 Resultado comparación distancias

Una vez realizada la prueba se pudo apreciar que en general los datos son más dispersos a medida que nos alejamos debido a que la potencia de luz del haz de infrarrojos se atenúa en el trayecto recorrido, pero que también a medida que nos alejamos aumenta la velocidad con que reconoce los 121 puntos de un rostro por lo que se decidió establecer la distancia 1,65 metros como punto base para una identificación del rostro, debido a que su relación menor velocidad en conjunto con menor desviación estándar la tiene este punto. Para mayores detalles de las pruebas realizadas ver Anexo A9.6.1.

6.3 Resultados de identificación obtenidos

Si bien es cierto existen las llamadas “características de forma única” (Shape Unit), que nos entregaron una pauta de que puntos analizar, el desarrollo de esta primera etapa de software fue hecho con las distancias entre los puntos más representativos y fiables encontrados, a lo largo de pruebas realizadas, es decir, distancias que eran únicas y que a la vez sus puntos tienen una estimación errónea más baja.

A continuación se muestran los resultados obtenidos con cada una de las técnicas que se utilizaron. Durante las pruebas se realizaron 5 intentos de autenticación y 5 intentos de suplantación. Este procedimiento se realiza con 5 personas. El total de personas registradas en la base de datos es de 20 personas. Dentro de las tablas se muestra la identificación lograda, que corresponde a los verdaderos positivos presentes en las pruebas realizadas, además se muestra las suplantaciones, que corresponden a los falsos positivos realizados por una persona diferente (no legítima), y para finalizar dentro de la tabla se muestra el porcentaje de aciertos totales entre verdadero positivo y verdadero negativo (Rechazar a un individuo que no es legítimo, complemento de falso positivo). Es importante destacar que las personas presentes en las siguientes tablas no son siempre las mismas, o se encuentran en un orden diferente.

6.3.1 Distancias

En este caso se utiliza una de las variantes de autenticación basada en rasgos locales. Se utilizan las distancias entre puntos característicos para poder determinar la identidad de la persona. Al realizar una captura se obtienen los patrones de la malla de puntos de la personas y estos se comparan con los de la base de datos.

Persona	Verdadero Positivo	%	Falso Positivo	%	Total
Persona 1	2	40	3	60	40
Persona 2	3	60	2	40	60
Persona 3	4	80	1	20	80
Persona 4	3	60	3	60	50
Persona 5	4	80	2	40	70
Promedio	3.2	64	2.2	44	60

Tabla 11 Distancias

6.3.2 Áreas

En este caso se utilizan las áreas triangulares formadas entre algunos puntos característicos para poder determinar la identidad de la persona. Al realizar una captura se obtienen los patrones de la malla de puntos de la personas y estos se comparan con los de la base de datos.

Para realizar el cálculo del área se utiliza la fórmula de Harón. Esta entrega el área de un triángulo dado el largo de los 3 lados. La fórmula es la siguiente:

$$A = \sqrt{s(s - a)(s - b)(s - c)}$$

Siendo a,b,c: los lados del triángulo y s el semiperímetro $s = \frac{a+b+c}{2}$

Persona	Verdadero Positivo	%	Falso Positivo	%	Total
Persona 1	4	80	3	60	60
Persona 2	3	60	2	40	60
Persona 3	2	40	3	60	40
Persona 4	4	80	1	20	80
Persona 5	2	40	4	80	30
Promedio	3	60	2.6	52	54

Tabla 12 Áreas

6.3.3 Distancias Normalizando Datos

Unas de las técnicas que se utilizara para mejorar los resultados es la normalización de los datos. La normalización consiste en llevar cada punto a un espacio vectorial diferente. Se divide cada vector (punto del plano 3D) por su norma, obteniendo como resultado otro vector. Esto se representa con la siguiente formulas.

$$U = V/\sqrt{(x^2 + y^2 + z^2)}$$

El vector V se divide por su normal. Siendo su norma la raíz de la suma de cada una de las coordenadas del vector al cuadrado.

La siguiente tabla muestra los resultados de la misma prueba de distancias anterior, pero con datos normalizados.

Persona	Verdadero Positivo	%	Falso Positivo	%	Total
Persona 1	2	40	3	60	40
Persona 2	3	60	4	80	40
Persona 3	4	80	2	40	70
Persona 4	3	60	4	80	40
Persona 5	3	60	3	60	50
Promedio	3	60	3.2	64	48

Tabla 13 Distancias Normalizando Datos

6.3.4 Áreas Normalizando Datos

Estas pruebas es igual a la realizada anteriormente con áreas triangulares, solo que aplicando normalización de los datos. En este caso se debe utilizar la fórmula de normalización de los datos y después la fórmula de Herón para el cálculo de áreas.

Persona	Verdadero Positivo	%	Falso Positivo	%	Total
Persona 1	4	80	3	60	60
Persona 2	1	20	1	20	50
Persona 3	4	80	4	80	50
Persona 4	2	40	4	80	30
Persona 5	3	60	3	60	50
Promedio	2.8	56	3	60	48

Tabla 14 Áreas Normalizando Datos

6.3.5 Distancias Aplicando PCA

Una de las técnicas más usada en la literatura es PCA, es por esto que se decidió incluir esta técnica y evaluar sus resultados. Para esto se utilizaron 20 personas para realizar el proceso de entrenamiento.

Los resultados con esta técnica no resultaron muy satisfactorios. Posiblemente se debe a que la gran dispersión de los datos entregados por el dispositivo Kinect, no permiten realizar una imagen promedio lo suficientemente representativa de la muestra utilizada. Los resultados obtenidos al realizar la prueba con distancias utilizando PCA son los siguientes:

Persona	Verdadero Positivo	%	Falso Positivo	%	Total
Persona 1	5	100	1	20	90
Persona 2	4	80	4	80	50
Persona 3	4	80	3	60	60
Persona 4	4	80	2	40	70
Persona 5	2	40	4	80	30
Promedio	3.8	76	2.8	56	60

Tabla 15 Distancias Aplicando PCA

6.3.6 Áreas Aplicando PCA

Al igual que en distancias, se incluyó PCA para calcular las diferencias de las áreas de los rostros ingresados, con el las áreas del rostro generado con PCA entregando los siguientes resultados

Persona	Verdadero Positivo	%	Falso Positivo	%	Total
Persona 1	3	60	3	60	50
Persona 2	3	60	4	80	40
Persona 3	4	80	1	20	80
Persona 4	3	60	3	60	50
Persona 5	5	100	1	20	90
Promedio	3.6	72	2.4	48	62

Tabla 16 Areas Aplicando PCA

6.3.7 Promedio Usando Distancias

El principal problema es la dispersión de los datos que entrega el dispositivo Kinect. Es por esto que se decidió minimizar esta dispersión. Cada captura realizada a los usuarios captura 36 “frames” de los puntos de los rostros, con el fin de ser promediados y obtener como resultado un único “frame”.

Otra medida para eliminar la dispersión de los datos fue, no considerar en el cálculo del promedio, los “frames” que tuvieran una mayor desviación estándar. De esta forma no se consideran algunas de las capturas que crean confusión en la identificación.

Al aplicar esta técnica y aplicarlas a los patrones dados por las distancias entre puntos, se obtuvieron los siguientes resultados.

Persona	Verdadero Positivo	%	Falso Positivo	%	Total
Persona 1	4	80	2	40	70
Persona 2	5	100	2	40	80
Persona 3	3	60	1	20	70
Persona 4	5	100	1	20	90
Persona 5	5	100	4	80	60
Promedio	4.4	88	2	40	74

Tabla 17 Promedio Usando Distancias

6.3.8 Promedio Usando Áreas

Los resultados de las distancias mejoraron al aplicar el promedio de los “frames”. Es por esto que se espera que los resultados con las áreas también se vea favorecido. A continuación se muestran los resultados al aplicar el mismo procedimiento usado anteriormente con distancias, aplicado en las áreas.

Persona	Verdadero positivo	%	Suplantaciones	%	Total
Persona 1	5	100	2	40	80

Persona 2	5	100	4	80	60
Persona 3	4	80	1	20	80
Persona 4	4	80	0	0	90
Persona 5	5	100	3	60	70
Promedio	4.6	92	2	40	76

Tabla 18 Promedio Usando Áreas

6.3.9 Promedio Usando Áreas y Distancias

La última prueba consiste en combinar los mejores patrones obtenidos de la prueba de distancias y áreas. Aplicando la misma técnica aplicada anteriormente a estas. A continuación se muestran los resultados al comparar utilizando patrones de ambas.

Persona	Verdadero positivo	%	Falso Positivo	%	Total
Persona 1	5	100	1	20	90
Persona 2	5	100	4	80	60
Persona 3	5	100	1	20	90
Persona 4	4	80	0	0	90
Persona 5	5	100	2	40	80
Promedio	4.8	96	1.6	32	82

Tabla 19 Promedio Usando Áreas y Distancias

7 Conclusiones

El reconocimiento facial es una forma de identificación que en la actualidad llama mucho la atención, ya que aseguraría que la persona que se presente para la autenticación es quien dice ser. De esta manera se eliminarían problemas de otras técnicas de identificación, como comúnmente pasa con el robo de claves o de tarjetas de acceso.

El reconocimiento facial es la tendencia de hoy en día, y por lo mismo es que existen una gran cantidad de aplicaciones, con distintas funcionalidades como el control de acceso, identificación automática de personas, seguridad de la información, seguridad en bases de datos, seguridad en internet y permisos en cajeros automáticos, etc.

Lograr identificar un rostro a través del dispositivo Kinect y que éste presente resultados aceptables, es una tarea compleja de realizar, esto debido principalmente a las limitaciones que posee el sensor, el cual no posee una fiabilidad de datos lo suficientemente aceptable. Es por esta razón que fue necesario establecer un escenario ideal, para un correcto desempeño de Kinect, eligiendo una distancia propicia para el reconocimiento facial, y así mitigar el error de precisión presente en este dispositivo.

Dentro de las primeras técnicas a evaluar se utilizaron técnicas de cálculos de áreas y distancias, las cuales no dieron resultados satisfactorios. Es por esto que se decidió recurrir a otros métodos, como la normalización de los datos, técnica que no mejoro los resultados obtenidos, debido a que no solo maximizo las diferencias entre personas, sino que también maximizo las diferencias de un mismo individuo. Es por esto que se decidió descartar la normalización de los datos para el prototipo final.

Otra de las técnicas evaluadas fue PCA, técnica que tampoco mejoro los resultados, esto debido principalmente a la gran dispersión de datos, sumado a que esta técnica no está orientada a un enfoque de puntos característicos en 3 dimensiones.

Los mejores resultados se dieron con la creación de una técnica la cual promediaba los “frames” de patrones con áreas y distancias y luego le restaba los “frames” que estuvieran más alejados del promedio obtenido. De esta forma fue posible disminuir la dispersión los datos presente en Kinect y obtener una tasa superior de aciertos.

Si bien el prototipo final logro incrementar la tasa de aciertos superando el 80% esperado, este número no representa un sistema robusto, ya que en temas de seguridad, la precisión es un factor clave.

En conclusión, el dispositivo Kinect 1.5 no cuenta con la precisión para poder llegar a buenos resultados, sin embargo el trabajo realizado logró acercarse a un prototipo de autenticación facial, que podría tener mejores resultados con un dispositivo con mayor exactitud, ya que el Sensor Kinect de Microsoft no fue diseñado para este propósito.

8 Referencias

- [1] Álvaro Ángel Orozco, Mauricio Álvarez, Juan Gabriel Fetecua, *Reconocimiento de expresiones faciales utilizando análisis de componentes principales kernel (kpca)*, Universidad Tecnológica de Pereira, pp. 43-47, 2008.
- [2] Cecilia Aguerrebere Otegui, Germán Capdehourat Longres, Mauricio Delbracio Bentancor, Matías Mateu Graside, *Proyecto Aguará Reconocimiento de Caras*, Facultad de Ingeniería Universidad de la República, pp. 11-51, 2006.
- [3] Ian Sommerville, *Ingeniería del Software*, Pearson Educación S.A, pp. 245-249, 2005.
- [4] Jennifer Huang, Bernd Heisele and Volker Blanz, *Component-based Face Recognition with 3D Morphable Models*, Center for Biological and Computational Learning, pp 2-7, 2008.
- [5] Roger Gimeno Hernández, *Estudio de Técnicas de Reconocimiento facial*, Universidad politécnica de Cataluña, pp. 15-44, 2010.
- [6] L. Wiskott, J.-M. Fellous, N. Krueger, C. von der Malsburg, *Face Recognition by Elastic Bunch Graph Matching*, IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 3-10, 1997.
- [7] Henry Arguello Fuentes, *Sistema de Reconocimiento Basado en la Imagen Facial*, Universidad Industrial de Santander, pp. 9-12, 2011.
- [8] Nicolas Lopez Perez, Juan José Toro Agudelo, *Técnicas de Biometría Basadas en Patrones Faciales del Ser Humano*, Universidad Tecnológica de Pereira, pp. 54-62, 2012.
- [9] Pedro Martins, Joana Sampaio, Jorge Batista, *Facial Expression Recognition Using Active Appearance Models*, University of Coimbra, pp. 3-7, 2009.
- [10] Ana Belén Moreno Díaz, *Reconocimiento Facial Automático mediante Técnicas de Visión Tridimensional*, Universidad Politécnica de Madrid, pp. 168-179, 2004.
- [11] Microsoft Corporation, sitio web oficial de Microsoft, <http://www.microsoft.com>.
- [12] S.A. Visión computacional e ingeniería avanzada, Sitio oficial de VisionLabs, <http://visionlabs.cl/blog/?p=148>.
- [14] Chenghua Xu, Yunhong Wang, Tieniu Tan, Long quan, *A New Attempt To Face Recognition Using 3d Eigenfaces*, Hong Kong University of Science and Technology, pp 2-5, 2004.
- [15] Roger Pressman, *“Ingeniería del Software: Un enfoque práctico”*, 5ta Edición, University of Connecticut, pp 36-38, 2013.
- [16] Timo Ahonen, Abdenour Hadid, *Face Recognition with Local Binary Patterns*, University of Oulu, Finland, pp 471-480, 2009.

9 Anexos

9.1 Conceptos

- **Autenticación** Es el acto de establecimiento o confirmación de algo o alguien como auténtico. En particular la autenticación de una persona consiste en verificar que la persona es quien dice ser.
- **Biometría**
La biometría es una ciencia que se dedica al estudio del cuerpo humano como medio de identificación y autenticación de algún sistema, pudiendo utilizar características físicas (huella dactilar, iris, geometría de la mano, rostro) y de comportamiento (voz, firma, dinámica del tecleo o forma de caminar).
- **Patrón**
Conjunto de características únicas, que son utilizadas para reconocer un objeto.
- **Software Development Kit (SDK):** Es el kit de desarrollo que provee Microsoft para la creación de aplicaciones que utilizan la tecnología Kinect. El SDK es una librería que permite controlar la captura de imágenes y audio capturados por el dispositivo Kinect.
- **Imagen:** Es la proyección en perspectiva en el plano bidimensional de una escena tridimensional en un determinado instante de tiempo.
- **Histograma (Histogram):** Es la representación gráfica de las frecuencias relativas con las que aparecen los distintos colores en una determinada imagen. Proporciona información sobre el brillo y el contraste de la imagen, y puede ser utilizado para ajustar estos parámetros, eliminar ciertas tonalidades molestas, etc...
- **Umbral (Threshold):** La forma más sencilla de obtener una imagen monocromática es fijar un nivel de gris que constituye el umbral que nos indica si el punto se pinta. Por ejemplo, si pintamos puntos blancos sobre fondo negro, una vez fijado el umbral, los puntos cuya intensidad lo sobrepase se pintarán de blanco, los demás no aparecerán, por lo tanto un umbral bajo producirá una imagen muy clara.
- **Falso Positivo:** Aceptar a un individuo que no es legítimo.
- **Falso Negativo:** Rechazar a un individuo que sea legítimo.
- **Verdadero Positivo:** Aceptar a un individuo que es legítimo.
- **Verdadero Negativo:** Rechazar a un individuo que no es legítimo.

9.2 Principales Técnicas Holísticas

9.2.1 Principal Component Analysis (PCA)

El análisis de componentes principales es una técnica de reducción dimensional, que permite encontrar los vectores que mejor representan la distribución de un grupo de imágenes [5].

PCA permite representar una imagen como una matriz de $N \times N$ y posteriormente en un vector de N^2 . Primero se obtiene una imagen promedio a través de la vectorización un conjunto de imágenes de rostros y obteniendo un promedio. Después se resta el rostro promedio a cada uno de los vectores de las imágenes, obteniendo un nuevo conjunto de vectores T con las diferencias con la cara promedio. De esta forma se obtiene el patrón que diferencia cada una de las imágenes. La reducción dimensional realizada por PCA es equivalente al número de autovectores que se utilicen. Por lo tanto la imagen proyectada por PCA tendrá una dimensión de valor d como lo muestra la figura A9-1.

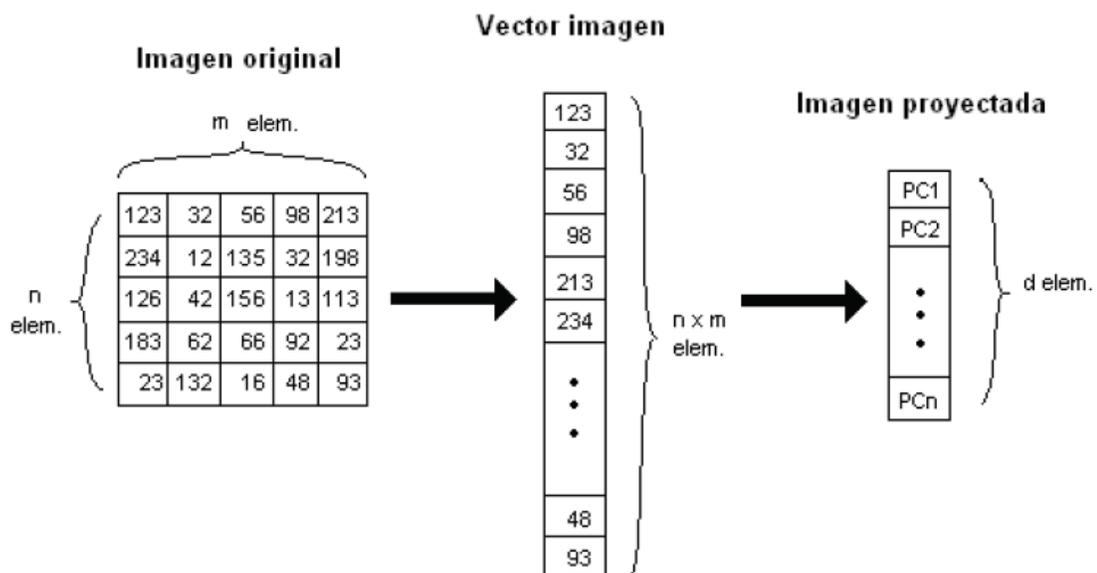


Figura A9-1 PCA [5]

Estos autovectores representan las componentes principales que son más comunes en imágenes de diferentes caras. Esto permite realizar una comparación de un patrón de nueva captura con los patrones de los vectores del conjunto T .

9.2.2 Análisis de discriminante lineal (LDA)

Este algoritmo tiene como finalidad convertir un problema de alta dimensión en uno de baja dimensión. Esto se consigue maximizando la varianza entre clases (entre usuarios) y minimizando la varianza al interior de una clase (de cada usuario) [5]. Esto quiere decir, que se clasifican los usuarios como clases, donde cada clase posee varias imágenes, la

varianza entra las clases se maximizan y la varianza al interior de las clases se minimiza (ver figura A9-2).



Figura A9-2 LDA [5]

LDA tiene problemas cuando se enfrenta a datos faciales de alta direccionalidad, muchas veces la muestra se hace muy pequeña, ya que no posee las suficientes imágenes de una clase para la correcta identificación de la imagen de entrada.

9.2.3 Análisis de Componentes Independientes (ICA)

Es una técnica desarrollada en el año 2002 por Bartlett, Movellan y Sejnowski. Este método aborda el problema de la representación de rostros en un espacio de baja direccionalidad desde un punto de vista estadístico. Esta técnica enfrenta el problema tratando a las imágenes o a los píxeles de cada imagen como una variable aleatoria respectivamente. ICA proyecta las imágenes de un espacio de baja dimensión formado por un conjunto de vectores estadísticamente independientes y usa la misma técnica que PCA para reducir la direccionalidad [7].

9.2.4 Métodos Kernel

Son métodos compatibles con los algoritmos recién nombrados, estos métodos son una alternativa para las diferentes tareas, como la eliminación del ruido (factores que no interesan en la captura de la imagen), reconocimiento de patrones y reducción de dimensionalidad. Básicamente el proceso kernel realiza un mapeo de los datos de entrada a otro espacio característico, esto permite una generalización no lineal del mayoría de los algoritmo lineales como algunos de los vistos anteriormente. La utilización métodos kernel pueden usarse en algunos casos para complementar las demás técnicas y da origen a técnicas como KPCA, KLDA, KICA, entre otros [1].

9.2.5 Evolutionary Pursuit (EP)

Esta técnica plantea una forma diferente de obtener una base de vectores para la representación de las imágenes de los rostros. Para obtener dicha base se realiza una búsqueda de manera de maximizar una función fitness que mide la precisión de la clasificación y la habilidad de generalización del sistema. Como el problema de buscar la base óptima es un problema de dimensión alta se utiliza un tipo de Algoritmos Genéticos (GA) al cual se le llama Evolutionary Pursuit [8].

EP consiste primeramente en reducir la dimensión de los datos mediante PCA, luego a este espacio vectorial se le aplica la transformación de whitening (transformación de variables aleatorias en otro grupo de variables cuya covarianza es la matriz identidad) de manera de encontrar una base óptima que sirva para reconocer los rostros.

9.2.6 Support Vector Machine (SVM)

Es una herramienta de carácter genérico que sirve para resolver problemas de reconocimiento de patrones. Dado un conjunto de puntos en un determinado espacio que pertenecen a dos clases distintas, SVM encuentra el hiperplano que separa la mayor cantidad de puntos de la misma clase del mismo lado. Esto se realiza maximizando la distancia de cada clase al hiperplano de decisión denominado OSH (Optimum Separating Hyperplane). Los puntos más cercanos al hiperplano, de cada conjunto en cuestión, son los llamados vectores soporte (support vectors) [10].

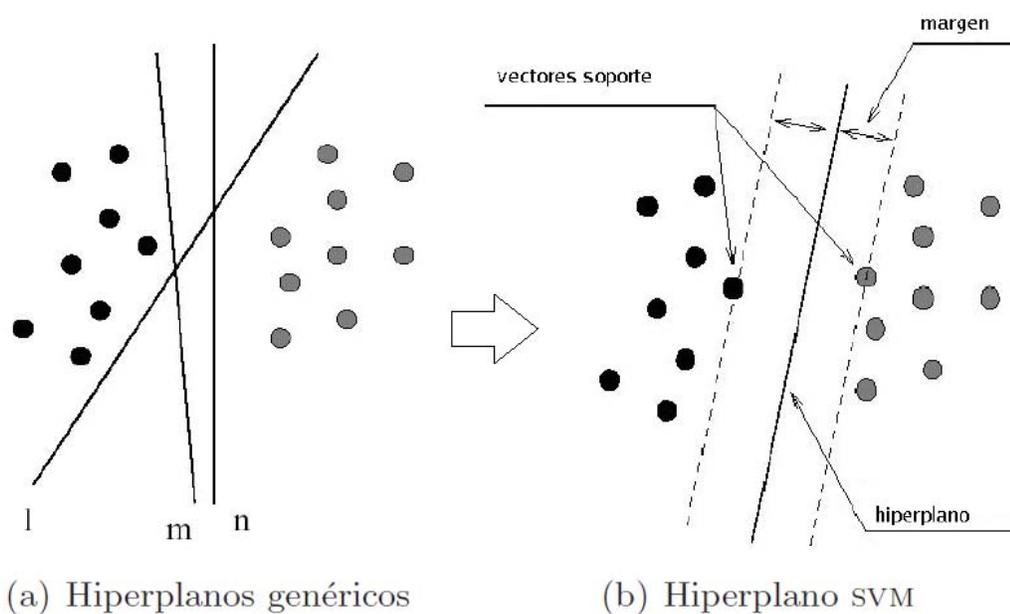


Figura A9-3 SVM [10]

9.3 Principales Técnicas Basadas en Rasgos Locales

9.3.1 AAM Active Appearance Model

AAM (Modelo Activo de Apariencia o Active Appearance Models) es un modelo estadístico de la estructura interna de un objeto cuyo contorno está marcado por puntos, estos puntos reciben el nombre de landmarks, este modelo se genera combinando modelos de variación de la forma y la apariencia de la cara en niveles de gris del objeto de interés, pudiendo generalizarse para la mayoría de los objetos.

Ajustar dicho modelo a una imagen implica encontrar los parámetros del modelo que generan una imagen sintética lo más cerca posible a la imagen objetivo y debido a la gran cantidad de parámetros presentes en el modelo hacen de éste un problema complicado.

Los modelos de forma y apariencia, se generan considerando distintos puntos marcados en la imagen como vectores y aplicando PCA a dicha información. Dado que existe correlación entre la variación de la forma y la apariencia, se aplica PCA a la concatenación de los vectores de los modelos anteriores y con ello se construye el modelo final [9].

Los tres pasos para generar el modelo en el sistema AAM son:

CAPTURA -> NORMALIZACIÓN -> ANÁLISIS ESTADÍSTICO

- Captura: Se define un número determinado de puntos sobre el contorno del objeto que interesa para determinar la forma.



Figura A9-4 Cara con Landmarks [9]

Luego se muestra una imagen usando antes una función de ajuste (warping function) y separando, en zonas a comparar (ver Figura A9-5) la textura mediante un esquema de triangulación.

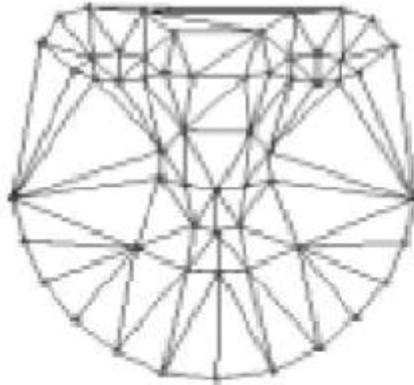


Figura A9-5 Partición de la textura en zonas [9]

- Normalización: Se busca un espacio (escenario) normalizado, para ello se alinean las formas eliminando la posición, escalado y orientación. Después se quita de la imagen los efectos de la iluminación, para así normalizarla, buscando un contraste y brillo estándar para todas las imágenes.
- Análisis estadístico: Se utiliza PCA para conseguir una descripción compacta y restrictiva, obteniéndose un vector de dimensión $2n$, diferente para cada una de las caras de nuestro conjunto de entrenamiento.

El sistema decisor compara los parámetros de la imagen capturada con la foto con los parámetros almacenados.

Dentro de las ventajas se destaca poder obtener representación de vistas no conocidas de la cara a partir de la interpolación de poses conocidas. Como aspecto negativo se menciona la complejidad en la implementación. Además los resultados son fuertemente sensibles a la correcta selección de los puntos principales.

9.3.2 Elastic Bunch Graph Matching (EBGM)

Este método consiste en extraer una representación de la cara en forma de grafo, con nodos situados en puntos de la cara, llamados “fiducial points ” y el reconocimiento se realiza comparando los grafos correspondientes a las distintas imágenes partiendo de la idea de que todas las caras comparten una estructura topológica similar.

Se define un conjunto de puntos principales, como por ejemplo las pupilas, esquinas de la boca, etc. Un grafo etiquetado representando una cara consiste en lo siguiente: N nodos ubicados en los puntos principales y las aristas que se forman entre parejas de nodos. Cada nodo es etiquetado con los denominados jets y cada arista es etiquetada con la distancia entre los nodos correspondientes.

Los jets se basan en una transformada wavelet definida como la convolución de la imagen con una familia de núcleos de Gabor con distintas frecuencias y orientaciones. Una representación de esto se muestra en la figura A9-6.

Para encontrar los puntos principales en nuevas caras se necesita una descripción general del objeto cara. Dicha descripción debe cubrir una amplia gama de variaciones como son diferentes formas de ojos, boca, diferentes sexos, razas, etc. Para ello se combina un grupo representativo de grafos modelo en una estructura del tipo stack llamada Face Bunch Graph (fbg). Cada grafo modelo tiene la misma estructura y los nodos refieren a los mismos puntos principales. La figura A9-7 muestra la representación gráfica del fbg. El objetivo en una imagen de prueba es encontrar los puntos principales y obtener el grafo que maximice la similitud con el fbg considerado. El grafo resultante es la representación de la cara presente en la imagen [6].

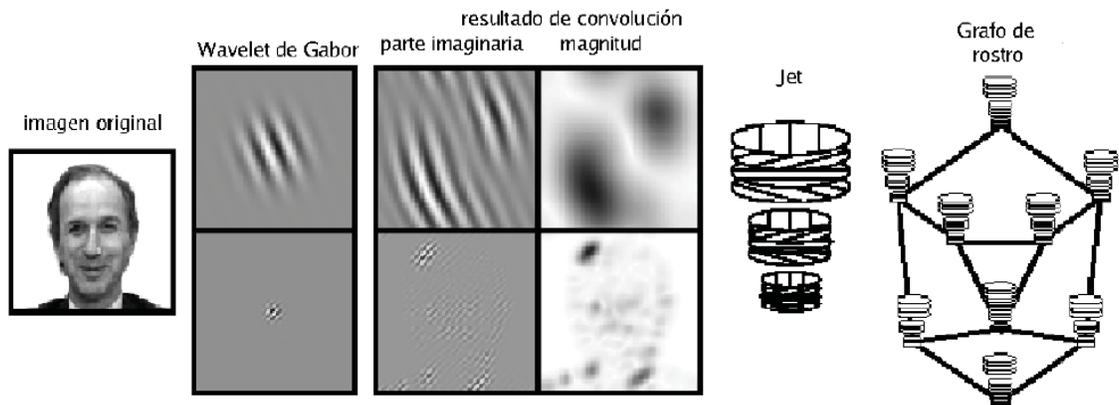


Figura A9-6 Representación de la cara en forma de grafo [6]

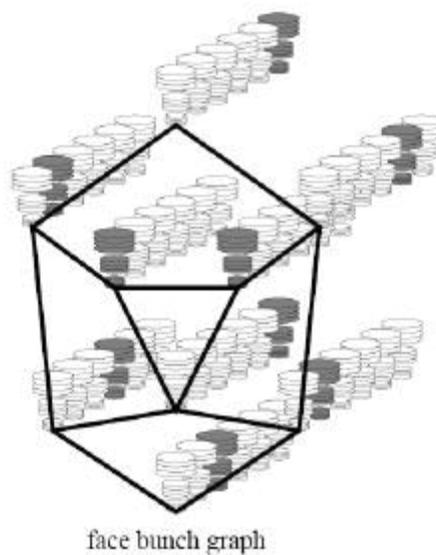
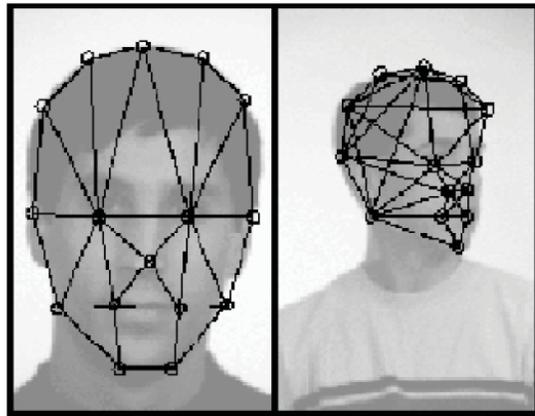
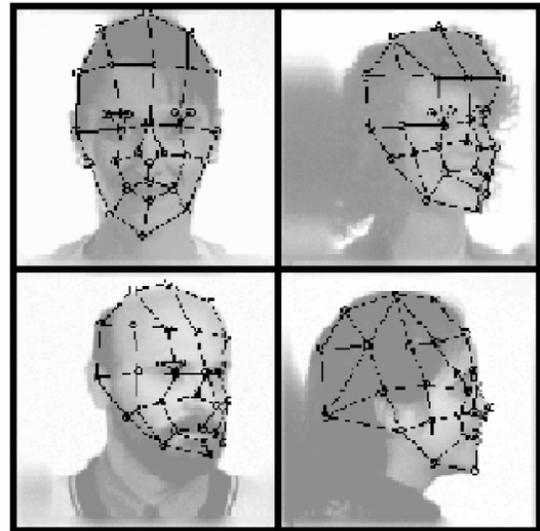


Figura A9-7 Representación de un FBG [6]



Grillas para detección del rostro



Grillas para reconocimiento de rostros

Figura A9-8 Ejemplo de grafos obtenidos utilizando EBGm [6]

A continuación se describe la sección del algoritmo EBGm que se refiere a la localización de características

Primero se define una estructura de grafo sobre la cara, cuyos nodos son puntos de interés, que se puedan localizar fácilmente y que posean la misma estructura en todos los rostros.

Luego cada nodo del grafo es caracterizado utilizando un banco de filtros de Gabor de diferentes frecuencias y orientaciones. En cada nodo, se calcula la respuesta de todos los filtros, a lo que se da el nombre de Jet. Por consiguiente, cada nodo queda etiquetado con sus coordenadas y su Jet asociado.

Finalmente tras capturar una imagen nueva se busca en la base de datos el conjunto de Jets que sean más similares. Para esto se centra el grafo en los ojos de la nueva imagen, se calculan los jets de estos puntos y se comparan contra los jets de los ojos en la base de datos. Para la comparación de los jets se utiliza una función de similitud por fase, ésta es similar a una correlación, y está definida como se describe en la ecuación 4.

$$S(J, J') = \frac{\sum_{j=1}^n a_j a'_j \cos(\phi_j - \phi'_j)}{\sqrt{\sum_{j=1}^n a_j^2 \sum_{j=1}^n a'^2_j}}$$

Dónde: a' es la magnitud del Jet de la base de datos, a la magnitud del Jet de la imagen nueva, θ' la fase del Jet de la base de datos, θ la fase del Jet de la imagen nueva y n el número de coeficientes del Jet.

9.3.3 Modelos de Markov escondidos (HMM)

Se trata de un conjunto de modelos estadísticos utilizados para caracterizar las propiedades estadísticas de una señal. Estos modelos son de gran utilidad para la representación de dependencias estadísticas en problemas que tienen una temporalidad inherente, por lo que han tenido éxito en aplicaciones como reconocimiento de voz y de gestos. Para el caso de imágenes, las dependencias temporales se transforman en dependencias espaciales, por lo que es necesario encontrar una forma de recorrer una señal que en vez de ser unidimensional, como es en el caso de la voz, sea de dos dimensiones [8].

Formalmente un HMM consiste en dos procesos interrelacionados:

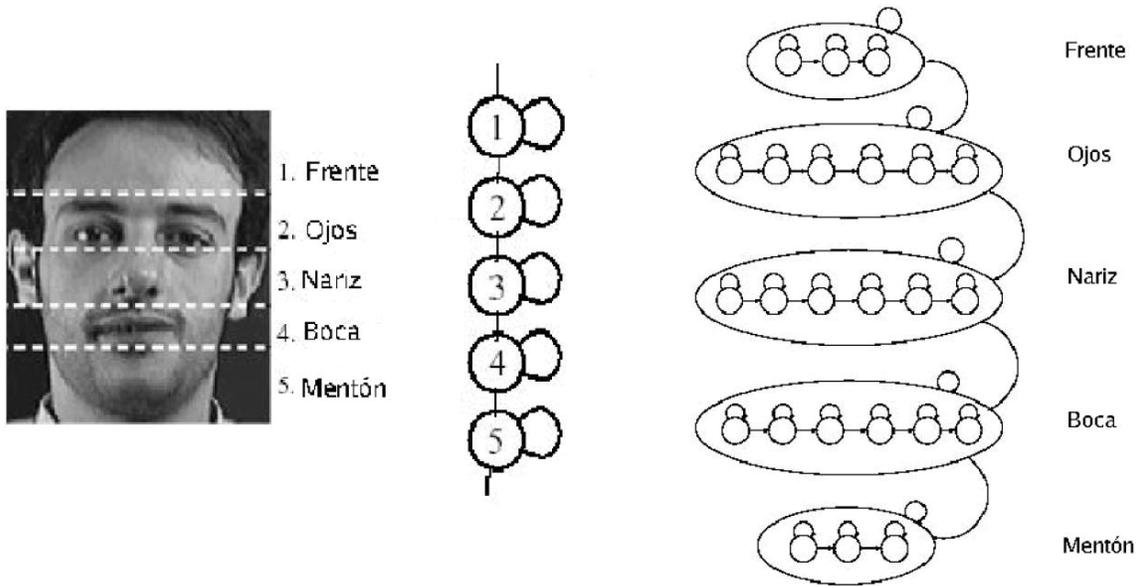
1. Un proceso subyacente, modelado por una cadena de Markov no observable con un número finito de estados, una matriz de transición de estados y una distribución de probabilidad de estados inicial.
2. Un proceso observable dado por los vectores de observación y relacionado por un conjunto de densidades de probabilidad asociadas a cada estado.

Para aplicar HMM al reconocimiento de caras, lo que se hace es dividir la imagen en bloques y se aplica la DCT o KLT. Luego ciertos coeficientes de la transformada del bloque se utilizan como los vectores de observación.

Se modela la cara por una cadena de Markov, la cual se entrena a los efectos de encontrar los parámetros, utilizando el algoritmo EM maximizando $P(O/\lambda)$, siendo O el vector de observación y λ la cadena de Markov. De esta forma se obtiene una cadena λ para cada individuo de la base. Luego para tomar la decisión se elige la cadena λ^* que maximice $P(O/\lambda_i)$. Existen distintos modelos para representar la cara, a los efectos de recorrer la señal bidimensional:

HMM top-down: Se toman bloques que se corresponden con las distintas regiones horizontales de la cara desde arriba hacia abajo. De esta manera los bloques corresponden a la frente, los ojos, la nariz, la boca y el mentón. En la figura A9-9(a) se puede observar esta idea.

HMM embebidas: Se hace una cadena de cadenas, donde una de nivel inferior recorre la imagen en sentido horizontal y luego una de nivel superior recorre la imagen en sentido vertical. Esta idea se puede ver en la figura A9-9(b).



(a) HMM top-down

(b) HMM embebidas

Figura A9-9 A la izquierda, división en franjas de un rostro para el algoritmo HMM. A la derecha, las cadenas de nivel inferior y superior para el algoritmo HMM [8]

Existen otras formas de modelar la cara, pero todas mantienen la idea de dividir la misma en bloques y tomar como vectores de observación algunos coeficientes de la dct o klt.

9.3.4 Local Binary Patterns (LVP)

Es una herramienta usada para describir texturas. Este recorre la imagen y etiqueta cada pixel, umbralizando una vecindad de 3x3 respecto al pixel en el que se encuentra, considerando el resultado como un numero binario. De esta forma se puede utilizar el histograma de las etiquetas como un descriptor de textura. En la figura A9-10 se muestra el funcionamiento de LVP [16]

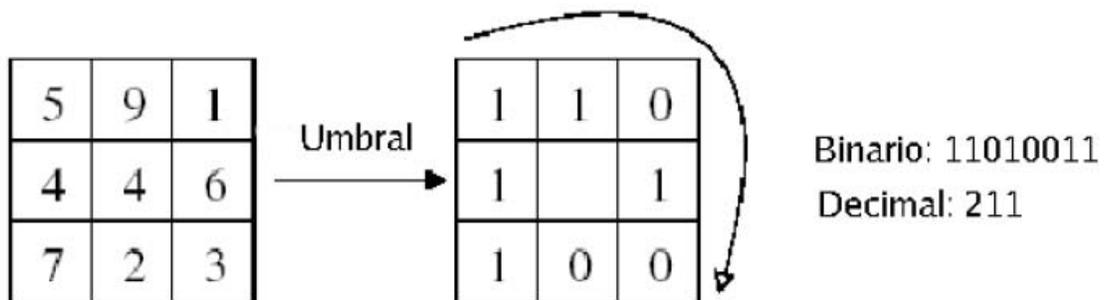


Figura A9-10 LVP [16]

Algunas variantes al operador básico fueron introducidas posteriormente. Una de ellas es considerar entornos de tamaño variable. Usando vecindades circulares e interpolando los valores de los píxeles en donde cualquier radio es válido. Otra variante es tomar en cuenta solamente los patrones uniformes, siendo estos los que tienen al menos dos transiciones 0-1 o 1-0. Esta idea se apoya en el hecho de que la mayor parte de los patrones en una imagen son de tipo uniforme.

El histograma obtenido con el anterior procedimiento contiene información acerca de la distribución local de los micropatrones, como bordes, puntos y otros. Para tener una representación eficiente de la cara es necesario además almacenar información espacial. Por lo tanto, la imagen se divide en regiones uniformes mediante una cuadrícula de bloque de 7x7. Luego se calculan los histogramas para cada bloque y se concatenan todos para obtener el patrón de la cara. De esta forma se logra una descripción de la cara en tres escalas o niveles: las etiquetas contienen información de los patrones a nivel de los píxeles, los histogramas en una pequeña región producen información a nivel regional y los histogramas regionales son concatenados para crear una descripción global de la cara.

9.3.5 Modelado 3D

La idea central de esta familia de técnicas es ajustar un modelo genérico 3D a cada imagen en cuestión. Para esto es necesario contar con imágenes 3D. Existen distintas técnicas de adquisición de imágenes (o reconstrucción) 3D, entre ellas: cámaras estéreo, escaners laser, SLS (Structure Light System), secuencia de imágenes 2D.

❖ Modelos Deformables: 3D Morphable Models:

Este sistema de reconocimiento de caras combina modelos deformables en 3D con una simulación gráfica de la proyección y la iluminación realizada por ordenador. Dada una imagen del usuario, el algoritmo automáticamente estima la forma, la textura y el resto de parámetros relevantes de la representación 3D. El modelo deformable de la cara, se basa en una representación vectorial de las caras. Este espacio se construye tal que, cualquier combinación convexa que pertenezca al espacio de S y T [4].

Para asegurar que los continuos cambios en a y b representen una transición de una cara a otra, se debe garantizar que no existe nada que coaccione la correspondencia punto a punto. Esto se hace mediante la generalización de la técnica de flujo óptico (al ser un modelo deformable, se usa para ajustar la máscara a la cara, en una secuencia de video), aplicada sobre imágenes tridimensionales en escala de grises. Los vectores S y T (***) se extraen directamente del modelo 3D, donde el vector S representa la concatenación de los ejes cartesianos (x, y, z) de los puntos; y el vector T es la concatenación de la textura (R, G, B). Se aplica el análisis PCA a los vectores S y T, usando el modelo de Phong para describir la reflexión difusa y especular de la superficie; con el fin de obtener un modelo deformable promedio. Mediante una función de coste, el algoritmo de ajuste (fitting) optimiza una serie de coeficientes de forma y coeficientes de textura, usando para ello 22 parámetros de rendering. Estos parámetros se recogen en un vector de características,

vector r , agrupando en él, datos relativos al ángulo de la pose, la traslación en 3D, la distancia focal, etc. Se emplean dos paradigmas para probar este método. En el primero de todos, se analizan toda la colección de imágenes con el algoritmo de ajuste (fitting), almacenando la forma y la textura. De la misma manera, para una imagen de prueba, se calculan todos los coeficientes y se compara con la colección de datos, obtenida anteriormente, para encontrar la mejor representación gráfica, como se muestra en Figura A9-11.

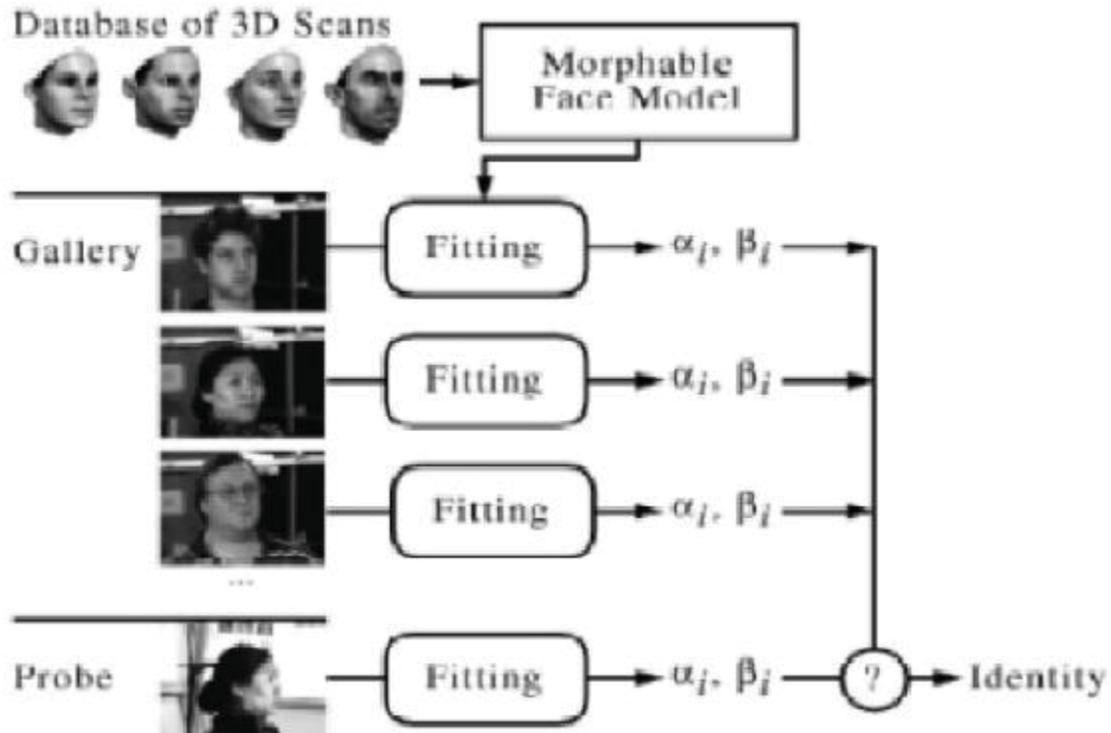


Figura A9-11 Para el reconocimiento, los coeficientes α y β de la imagen de prueba, se comparan con los coeficientes de la colección de imágenes de entrenamiento [4]

❖ 3D Eigenfaces

Este método, se aplica sobre los modelos de cara de la base de datos, en la cual los modelos son representados por una nube de puntos (ver Figura A9-12).



Figura A9-12 Modelo de Cara representado por una nube de puntos [14]

El primer problema que hay que mencionar consiste en construir la nube de puntos. Esto se hace mediante un algoritmo iterativo donde lo primero, es localizar la punta de la nariz, que corresponde con el punto más prominente de la nube de puntos, entonces, se alinea el punto encontrado con una red básica. Esta red, se subdivide en cada iteración, como se muestra en la Figura A9-13.

Se considera el proceso suficientemente preciso con cuatro niveles.

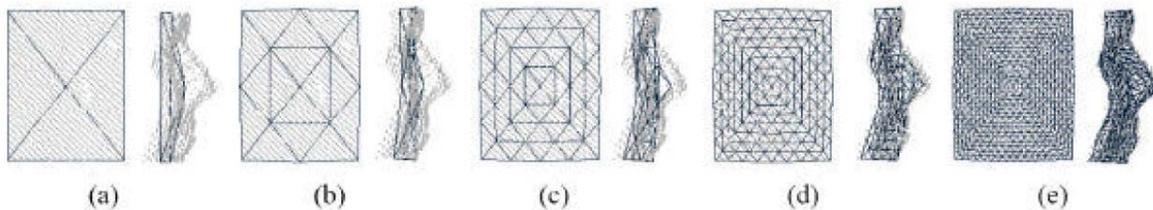


Figura A9-13 La red modelada en diferentes niveles. (a) red Básica. (b) de Primer Nivel (c) de Segundo Nivel. (d) de Tercer Nivel. (e) de Cuarto Nivel [14]

Las nubes de puntos tienen diferentes orientaciones, conservando las redes resultantes esta orientación. Se calcula un modelo medio y se alinean a este modelo, todas las redes, existiendo para ello, seis parámetros para las rotaciones y seis para las traslaciones.

Debido al ruido, a veces, no se pueden construir algunos modelos de red que permitan describir la forma geométrica del individuo. Estos modelos se denominan modelos de no caras.

Cada red contiene 545 nodos (puntos), donde los puntos son usados como una imagen de intensidad bidimensional, en donde la intensidad del píxel es la tercera coordenada (coordenada z) de cada nodo. A estas imágenes de intensidad se les aplica la técnica eigenfaces o más conocida como PCA [14].

9.4 Descripción de la API

9.4.1 IFTFaceTracking

El interfaz principal es IFTFaceTracker, y se puede instanciar mediante el método FTCreateFaceTracker.

```
this.faceTrackerInteropPtr =
NativeMethods.FTCreateFaceTracker(IntPtr.Zero); if
(this.faceTrackerInteropPtr == null)
{
throw new InsufficientMemoryException("Cannot create face tracker.");
}
```

Tras su inicialización, permite seguir un rostro de forma síncrona enviando imágenes de color y profundidad mediante el uso del objeto IFTImage. Estos resultados son devueltos por medio de una instancia de IFTResult.

IFTFaceTracker proporciona el método CreateFTResult para crear una instancia de IFTResult en la que se almacenen los resultados específicos para el modelo que está empleando la instancia de IFTFaceTracker. Una aplicación necesitará crear una instancia de IFTResult para poder comenzar el seguimiento facial, y puede usar el método de IFTFaceTracker para obtener una matriz de áreas que potencialmente contengan caras a partir de los datos de imagen.

```
internal FaceTrackFrame CreateResult(out int hr)
{
IFTResult faceTrackResultPtr;
FaceTrackFrame faceTrackFrame = null;
this.CheckPtrAndThrow();
hr = this.faceTrackerInteropPtr.CreateFTResult(out faceTrackResultPtr);
if (faceTrackResultPtr != null)
{
faceTrackFrame = new FaceTrackFrame(faceTrackResultPtr, this);
}
return faceTrackFrame;
}
```

Para comenzar el seguimiento de una cara se necesita llamar al método StartTracking. Éste es un método sencillo que busca en la imagen de entrada una cara, determina su orientación y comienza a seguirla.

Una vez StartTracking ha comenzado a seguir un rostro de la forma indicada por IFTRResult, la aplicación deberá continuar el seguimiento facial mediante llamadas sucesivas a ContinueTracking, que usa información de las llamadas anteriores a StartTracking o ContinueTracking para mantener un seguimiento facial preciso.

Se debe seguir llamando a ContinueTracking hasta que se desee terminar el seguimiento facial o éste haya fallado, por ejemplo debido a que la persona cuya cara se estaba siguiendo salga fuera de la imagen de la cámara. Un fallo en el seguimiento facial es indicado mediante el estado de IFTRResult. Para comenzar de nuevo el seguimiento facial, la aplicación llama a StartTracking de nuevo y después a ContinueTracking de forma secuencial.

```
//Determina si ya se estaba realizando un seguimiento facial
this.startOrContinueTrackingStopwatch.Start();

if (this.trackSucceeded) //Se debe continuar el seguimiento facial
{
    hr = this.faceTrackerInteropPtr.ContinueTracking(ref
    faceTrackSensorData, headPointsObj, this.frame.ResultPtr);
}

else//Se debe iniciar un nuevo seguimiento facial
{
    hr = this.faceTrackerInteropPtr.StartTracking(
    ref faceTrackSensorData, ref regionOfInterest, headPointsObj,
    this.frame.ResultPtr);
}

this.startOrContinueTrackingStopwatch.Stop();

this.trackSucceeded = hr == (int)ErrorCode.Success && this.frame.Status
== ErrorCode.Success;

this.trackStopwatch.Stop();
```

ContinueTracking es una función relativamente rápida que usa la información existente del estado de una cara seguida, por ello es mucho más eficiente que StartTracking, que debe generar la información facial por completo. En algunas ocasiones es necesario llamar continuamente a StartTracking, especialmente si la aplicación tiene una tasa de imágenes baja o si la cara se mueve demasiado rápido.

También se pueden recuperar y editar Shape Units (SUs) mediante los métodos GetShapeUnits y SetShapeUnits. Las Shape Units sirven para determinar las medidas de ciertas características del rostro, lo que facilita la tarea de seguir rostros ya conocidos mediante el uso del método ContinueTracking.

9.4.2 IFTResult

La interfaz IFTResult proporciona acceso a los datos obtenidos como resultado de las llamadas de seguimiento facial realizadas mediante los métodos IFTFaceTracker.StartTracking e IFTFaceTracker.ContinueTracking. IFTResult se crea llamando al método IFTFaceTracker.CreateFTResult. IFTFaceTracker proporciona el método CreateFTResult ya que los resultados están relacionados con el modelo subyacente con el que IFTFaceTracker ha sido inicializado, y por lo tanto su creación depende en gran medida de éste.

Sobre una llamada exitosa de seguimiento facial, IFTResult proporciona acceso a la siguiente información:

- GetFaceRect: Un rectángulo expresado en función de las coordenadas de la imagen devuelta por la cámara que rodea la cara que se está siguiendo.
- Get2DShapePoints: Coordenadas 2D de los puntos clave sobre la cara alineada en coordenadas de imagen de vídeo. Incluye los 87 puntos 2D indicados en la siguiente imagen (además de otros 13 no representados).

9.4.3 IFTImage

Los datos de imagen (video y profundidad) para el seguimiento facial se proporcionan mediante FT_SENSOR_DATA, una estructura de datos que contiene punteros a una imagen de vídeo y a una imagen de profundidad. Se debe usar el método FTCreateImage para crear instancias de IFTImage.

IFTImage encapsula los datos de una imagen usada en seguimiento facial y especifica los formatos de imagen soportados. Además, proporciona buffer para guardar la imagen (Allocate) y también permite realizar un almacenamiento externo para ésta en disco (Attach). En el primer caso (Allocate) IFTImage libera la memoria asignada cuando se llama al método Reset. En el caso de Attach el programador es responsable de gestionar la memoria para indicar cómo guardar la imagen.

9.4.4 IFTModel

La interfaz IFTModel proporciona un sistema para convertir los puntos de seguimiento facial en una malla de vértices 3D en el espacio de cámara, que se superponen sobre el rostro de la persona que está siendo seguida. Su instancia es devuelta por el método

IFTFaceTracker.GetFaceModel. Esta interfaz proporciona múltiples métodos para obtener varias propiedades del modelo:

- GetSUCount, GetAUCount: Devuelve el número de Shape Units (SUs) o Animation Units (AUs) usadas en el modelo lineal 3D.
- GetTriangles: Devuelve los triángulos expresados mediante sus vértices en coordenadas 3D de la malla del modelo facial. Cada triángulo tiene el índice de sus tres vértices listado en sentido horario.
- GetVertexCount: Devuelve el número de vértices existentes en la malla del modelo 3D.

Además, IFTModel proporciona dos métodos para obtener un modelo de cara 3D en el espacio de vídeo de la cámara o proyectado en el plano de vídeo de la cámara. Estos métodos son:

- Get3DShape: Devuelve los vértices del modelo de cara 3D transformados mediante las SUs, AUs, la escala de dimensionado, rotación y translación enviados.
- GetProjectedShape: Devuelve los vértices del modelo de cara 3D transformados mediante las SUs, AUs, la escala de dimensionado, rotación y translación enviados, y los proyecta sobre la imagen de vídeo. Es especialmente útil a la hora de superponer la malla 3D o los puntos faciales detectados sobre el rostro de la persona seguida y poder cotejar los datos.

9.5 Algoritmo para la extracción de coordenadas

```
EnumIndexableCollection<FeaturePoint, Vector3DF> facePoints3D =  
frame.Get3DShape();  
int index = 0;  
foreach (Vector3DF vector in facePoints3D)  
{  
    Debug.WriteLine(string.Format("{0}: ({1}, {2}, {3})", index++,  
vector.X, vector.Y, vector.Z));  
}
```

9.6 Resultados

9.6.1 Problemas y limitaciones observadas

9.6.1.1 Resultados a 1,2 metros del dispositivo Kinect

Alumno 1

	1	2	3	4	5	Promedio	Desviación estándar
13-65	0,189441 34	0, 184509 31	0,1 8839754	0,1 8692027	0, 1858856 9	0,1870308 3	0,00175272
2-49	0,1 0985716	0, 108670 42	0,1 0902549	0,1 0948767	0, 1074795 1	0,1089040 5	0,00081825
50-48	0,0 5950257	0, 059445 7	0,0 5942064	0,0 5817203	0, 0561363 1	0,0585354 5	0,00129887
16-53	0,1 0087051	0, 101379 84	0,1 0121319	0,1 0082595	0, 1014455 1	0,101147	0,00025582
Desviación Estándar Promedio							0,00412566

Tabla 20 Alumno 2 (1.2 mts)

Alumno 2

	1	2	3	4	5	Promedio	Desviación estándar
13-65	0,192858 22	0, 188820 84	0,1 902004	0,1 8872065	0, 1896197 4	0,1900439 7	0,00150809
2-49	0,1 1061623	0, 111430 59	0,1 1241341	0,1 1213654	0, 1116174 6	0,1116428 4	0,00062244
50-48	0,0 5810644	0, 060118 4	0,0 5963868	0,0 615168	0, 0605762 2	0,0599913 1	0,00112805
16-53	0,1 0037684	0, 100378 98	0,1 0000921	0,1 0034377	0, 1003804 9	0,1002978 6	0,00014496
Desviación Estándar Promedio							0,00340354

Tabla 21 Alumno 1 (1.2 mts)

9.6.1.2 Resultados a 1,5 metros del dispositivo Kinect

Alumno 1

	1	2	3	4	5	Promedio	Desviación estándar
13-65	0,189288 79	0, 186998 61	0,1 881701	0,1 8783036	0, 1882474 9	0,1881070 7	0,0007383
2-49	0,1 0867873	0, 109397 76	0,1 0933921	0,1 0799585	0, 1089785 6	0,1088780 2	0,00051203
50-48	0,0 577286	0, 058988 63	0,0 5845956	0,0 5646239	0, 0572483 5	0,0577775 1	0,00088814
16-53	0,1 0111334	0, 100985 81	0,1 0093185	0,1 0123777	0, 1009041 6	0,1010345 9	0,00012444
Desviación Estándar Promedio							0,0022629

Tabla 22 Alumno 1 (1.5 mts)

Alumno 2

	1	2	3	4	5	Promedio	Desviación estándar
13-65	0,189859 9	0, 190289 15	0,1 8941683	0,1 8510124	0, 1887190 3	0,1886772 3	0,00186191
2-49	0,1 1120047	0, 111084 06	0,1 1065414	0,1 0721185	0, 1107427 8	0,1101786 6	0,00149734
50-48	0,0 5934263	0, 057659 88	0,0 5836524	0,0 5589921	0, 0585373 1	0,0579608 5	0,00116156
16-53	0,1 0034819	0, 100148 29	0,1 0040039	0,1 0154727	0, 1003935 1	0,1005675 3	0,00049839
Desviación Estándar Promedio							0,0050192

Tabla 23 Alumno 2 (1.5 mts)

9.6.1.3 Resultados a 1,65 metros del dispositivo Kinect

Alumno 1

	1	2	3	4	5	Promedio	Desviación estándar
13-65	0,187771 85	0, 187707 61	0,1 8760567	0,1 876615	0, 188958	0,1879409 3	0,00051146
2-49	0,1 0751269	0, 106562 19	0,1 0772618	0,1 0801363	0, 1079677 3	0,1075564 9	0,00052867
50-48	0,0 5611318	0, 054837 11	0,0 5669972	0,0 5744267	0, 0583102 9	0,0566805 9	0,00117961
16-53	0,1 0142483	0, 101724 34	0,1 0141077	0,1 0138557	0, 1015454 7	0,1014981 9	0,00012577
Desviación Estándar Promedio							0,0023455

Tabla 24 Alumno 1 (1.65 mts)

Alumno 2

	1	2	3	4	5	Promedio	Desviación estándar
13-65	0,190501 07	0, 188191 91	0,1 8560169	0,1 8612926	0, 1828650 1	0,1866577 9	0,00256532
2-49	0,1 113922	0, 113601 65	0,1 1123192	0,1 122451	0, 1093283 8	0,1115598 5	0,00139668
50-48	0,0 5926732	0, 061203 33	0,0 6009848	0,0 6190781	0, 0569322	0,0598818 3	0,00173066
16-53	0,1 0027361	0, 099900 76	0,1 0044317	0,1 0036275	0, 1007051 9	0,1003371	0,00026147
Desviación Estándar Promedio							0,00595412

Tabla 25 Alumno 2 (1.65 mts)

9.6.1.4 Resultados a 1,8 metros del dispositivo Kinect

Alumno 1

	1	2	3	4	5	Promedio	Desviación estándar
13-65	0,1895360 2	0, 190344 8	0,1 8790666	0,1 8922364	0, 1890196 2	0,1892061 5	0,00079109
2-49	0,1 0598035	0, 109475 66	0,1 0642852	0,1 0620781	0, 1066759 1	0,1069536 5	0,00128195
50-48	0,0 5437265	0, 057087 56	0,0 5525325	0,0 5419086	0, 0540197 1	0,0549848 1	0,00113403
16-53	0,1 019833	0, 100667 33	0,1 0187359	0,1 0181524	0, 1015171 5	0,1015713 2	0,00047764
Desviación Estándar Promedio							0,00368472

Tabla 26 Alumno 1 (1.8 mts)

Alumno 2

	1	2	3	4	5	Promedio	Desviación estándar
13-65	0,189293 3	0, 186765 1	0,1 9008729	0,1 8709695	0, 1886544 2	0,1883794 1	0,00127109
2-49	0,1 1093579	0, 103599 06	0,1 1227989	0,1 1216179	0, 1110131 5	0,1099979 4	0,00324793
50-48	0,0 5854423	0, 051631 67	0,0 6126039	0,0 6072307	0, 0588078 8	0,0581934 5	0,00344565
16-53	0,1 0032623	0, 103070 98	0,1 002663	0,1 002298	0, 1003369 5	0,1008460 5	0,00111316
Desviación Estándar Promedio							0,00907784

Tabla 27 Alumno 2 (1.8mts)