



Universidad de las Ciencias Informáticas
Facultad 1

*Componente para la detección de vitalidad durante el proceso de
reconocimiento de personas mediante rasgos faciales.*

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN
CIENCIAS INFORMÁTICAS.

Autor: *Sergio Lecier Soler González*

Tutores: *Lic. Yadier Perdomo Cuevas
Ing. Royli Hernández Delgado*

La Habana, junio de 2015

Declaración jurada de autoría

Declaro que soy el autor de la presente investigación titulada *Componente para la detección de vitalidad durante el proceso de reconocimiento de personas mediante rasgos faciales*, para optar por el título de Ingeniero en Ciencias Informáticas.

El presente trabajo fue desarrollado en el período 2014-2015.

Finalmente declaro que todo lo anteriormente expuesto se ajusta a la verdad, y asumo la responsabilidad moral y jurídica que se derive de este juramento profesional. Autorizo, como único autor, a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente declaración jurada de autoría en La Habana a los 18 días del mes de junio del año 2015.

Autor
Sergio Lecier Soler González

Tutor
Lic. Yadier Perdomo Cuevas

Tutor
Ing. Royli Hernández Delgado

Agradecimientos

A mi madre por su amor incondicional, por tantos sacrificios, por ser un ejemplo para mí, por su fuerza y carácter.

A mi hermano Damián por cuidar de mi madre cuando yo no estaba presente.

A mi abuela Juana, por ser también una madre para mí, por sobreponerse a la distancia y estar siempre junto a nosotros.

A mi novia Azalea, por ser una parte innegable de mi vida, por regalarme su amor, por estar ahí en los momentos buenos y malos, por su comprensión y apoyo a lo largo de estos años.

A mi padre, por su ayuda y confianza cuando la necesité.

A mi familia toda, por el apoyo que me han ofrecido en cada momento.

A todos los profesores que contribuyeron a mi formación como profesional, en especial a mis tutores y amigos Yadier Perdomo Cuevas y Royli Hernandez Delgado, a Denis Luis García Ruibal y Francisco Javier Hernández Cao por confiar en mí cuando la UCI salía de mi vida antes de lo previsto, a Joel y a Edistio por su maestría en el aula, al profe Sergio Reyes por tener siempre una palabra de aliento para mí, a Adrián Hernández por su tiempo, preocupación y oportunas sugerencias.

A mis compañeros de estudio, amigos en los tiempos buenos y malos a lo largo de la carrera, Herbert, Nilberto, Alexis, Rafael, Ana María, Anais y Saymel.

A mis compañeros de trabajo durante los últimos cuatro años, Cariheily, Luis Castro, Julio, Eduardo y Roque, por su apoyo y amistad.

A todos aquellos que de una forma u otra me ayudaron en la realización de este trabajo.

Dedicatoria

A la memoria de Eloy, que significó ser un padre para mí y siempre quiso que viniera a la universidad.

A mi madre, quien ha sido el pilar fundamental de este logro en mi vida.

Resumen

Los sistemas de reconocimiento de personas se caracterizan por utilizar diferentes elementos para la identificación unívoca de un individuo, aquellos que se basan en el análisis de los rasgos faciales disponen cada día de más participación en la sociedad moderna. El incremento de su uso ha traído como consecuencia que las técnicas para suplantar la identidad de las personas autorizadas sean más sofisticadas y los fabricantes de estos sistemas están inmersos en una lucha por hacer más confiables sus productos. Una de las formas más comunes de fraude la constituye el uso de imágenes previamente capturadas, ya sean fotos o videos que contengan la cantidad suficiente de rasgos faciales genuinos como para hacer que estos sistemas determinen la presencia de una persona autorizada frente a la cámara. La disponibilidad de un mecanismo que determine la ocurrencia de este tipo de fraude es de vital importancia a la hora de evaluar el nivel de confiabilidad que ofrece un sistema de reconocimiento.

En el Centro de Identificación y Seguridad Digital (CISED) de la Universidad de las Ciencias Informáticas (UCI) se desarrolló un Sistema de Reconocimiento Facial para ser utilizado en sus soluciones informáticas. Este sistema no ostenta entre sus potencialidades un mecanismo para la detección de vitalidad. Este trabajo propone una solución que al ser incluida en dicho sistema permita detectar la ocurrencia de incidentes de suplantación de identidad en el proceso de identificación de una persona.

El documento recoge los resultados de la investigación realizada, haciendo un estudio de los principales enfoques existentes para la detección de este tipo de fraude. Se explica la arquitectura y el diseño del sistema propuesto. Se describen las herramientas y tecnologías utilizadas, así como los artefactos generados en el proceso de desarrollo.

Palabras clave: sistemas de reconocimiento de personas, identificación, rasgos faciales, detección de vitalidad, suplantación de identidad.

Índice General

Índice de figuras	VIII
Índice de tablas	IX
Introducción	1
Campo de Acción	4
Métodos Teóricos	4
Métodos Empíricos	5
Justificación de la Investigación	5
Estructura del documento	6
1. Fundamentación teórica	7
1.1. Introducción	7
1.2. Algoritmos para la detección de vitalidad	7
1.2.1. Método basado en el análisis de la textura y la frecuencia	7
1.2.2. Método basado en el análisis del movimiento de los ojos	10
1.2.3. Método basado en el análisis del enfoque variable	11
1.2.4. Método basado en el pestañeo de los ojos	11
1.2.5. Método basado en la clasificación binaria	13
1.2.6. Método basado en el componente dependiente	15
1.2.7. Método basado en el movimiento de los labios.	17
1.2.8. Análisis basado en la combinación de técnicas.	18
1.3. Análisis de las investigaciones consultadas.	18
1.4. Metodología de desarrollo de software.	20
1.4.1. Metodologías ágiles.	20
1.5. Herramientas y tecnologías para la solución del problema.	22
1.5.1. Lenguaje de modelado.	22
1.5.2. Herramienta CASE.	23
1.5.3. Lenguaje de programación.	23

1.5.4. Entorno de desarrollo integrado (IDE).	24
1.5.5. Procesamiento digital de imágenes.	25
1.6. Conclusiones parciales.	25
2. Características del sistema.	27
2.1. Introducción	27
2.2. Propuesta de solución	27
2.2.1. Método basado en el análisis del movimiento de los ojos	27
2.2.2. Método basado en el análisis del enfoque variable	29
2.3. Modelo de Dominio	32
2.3.1. Glosario de conceptos del modelo de dominio	32
2.4. Principales funcionalidades	34
2.5. Requisitos no funcionales	35
2.6. Historias de usuario	36
2.7. Planificación	37
2.7.1. Plan de entrega	38
2.7.2. Plan de iteraciones	38
2.8. Diseño	39
2.8.1. Tarjetas CRC	39
2.8.2. Diagrama de clases del diseño	40
2.8.3. Patrones de diseño	42
2.9. Arquitectura	43
2.9.1. Patrón arquitectónico	44
2.10. Conclusiones parciales	44
3. Implementación y pruebas.	46
3.1. Introducción	46
3.2. Tareas de ingeniería	46
3.3. Estándares de codificación	48
3.3.1. Estilos para la capitalización de los identificadores	49

3.3.2. Comprensión y legibilidad del código	49
3.4. Diagrama de componentes	49
3.4.1. Descripción del diagrama de componentes	49
3.5. Método para la detección de movimiento en los ojos	51
3.5.1. Definición del umbral de decisión	51
3.5.2. Calcular el umbral de comparación	51
3.6. Método para la detección de volumen	53
3.6.1. Calcular el umbral de comparación	53
3.7. Pruebas	53
3.7.1. Pruebas unitarias	54
3.7.2. Pruebas de aceptación	54
3.7.3. Pruebas de rendimiento	58
3.8. Conclusiones parciales	59
Conclusiones generales	60
Recomendaciones	61
Referencias bibliográficas	62
Bibliografía consultada	67
A. Historias de usuario	68
B. Tarjetas CRC	73
C. Tareas de ingeniería	74
D. Resultado de las pruebas unitarias	78
E. Casos de prueba alfa	79

Índice de figuras

1.1. Extracción de las características de frecuencia.	8
1.2. Vector característico basado en la extracción LBP (a)Imagen facial original (b)Imagen con LBP aplicado (c)Histograma del código LBP de la imagen.	9
1.3. Estructura gráfica del pestañeo de ojos basado en el modelo CRF.	12
1.4. El flujo del análisis realizado en el enfoque propuesto, O representa un operador de características.	16
1.5. Flujo de trabajo en XP.	21
2.1. Binarización de imágenes falsas en (a) y en imágenes reales en (b).	28
2.2. Flujo para la clasificación de las imágenes.	28
2.3. Aplicando Suma Modificada de Laplace sobre imágenes.	30
2.4. Selección de regiones en (a) un rostro real y en (b) una foto.	30
2.5. Modelo de dominio.	33
2.6. Diagrama de clases del diseño.	41
2.7. Arquitectura del componente.	44
3.1. Diagrama de componentes.	50
3.2. Resultados de la definición del umbral de movimiento.	53
3.3. Resultados de la definición del umbral de enfoque.	54
3.4. Función para binarizar una imagen.	55
3.5. Función para calcular la distancia de hamming.	55
3.6. Resultados de las pruebas de rendimiento.	59
D.1. Función para realzar el grado de enfoque.	78
D.2. Función para calcular el grado de enfoque.	78

Índice de tablas

1.1. Detección rostro vivo para los métodos comparados.	13
1.2. Detección de fraude para los métodos comparados.	13
1.3. Comportamiento de [1] utilizando NUAA Imposter Database.	15
1.4. Comportamiento de [1] aplicando la extensión sugerida por Peixoto en [2].	15
1.5. Comportamiento al ejecutar las pruebas utilizando NUAA Database.	16
1.6. Comportamiento al ejecutar las pruebas utilizando PRINT-ATTACK Database.	17
1.7. Comportamiento al ejecutar las pruebas utilizando CASIA Database.	17
1.8. Ventajas y desventajas de los enfoques para la detección de vitalidad.	19
2.1. HU Detectar el grado de movimiento en la región de los ojos.	36
2.2. HU Detectar el grado de enfoque existente en imágenes consecutivas.	37
2.3. Plan de entrega.	38
2.4. Plan de iteraciones.	38
2.5. Tarjeta CRC Detector de Vitalidad.	39
2.6. Tarjeta CRC Proveedor de Características.	40
3.1. Tareas de ingeniería de la segunda iteración.	46
3.2. CP Detectar movimiento en la región del ojo.	56
3.3. CP Detectar el grado de enfoque existente en imágenes consecutivas.	56
3.4. CP Verificar la integración del componente con el Sistema de Reconocimiento Facial (SRF).	57
A.1. HU Detectar un rostro a partir de una imagen.	68
A.2. HU Extraer la región de los ojos.	68
A.3. HU Extraer la región de la nariz.	69
A.4. HU Extraer la región de la oreja.	69
A.5. HU Realizar el proceso de binarizado sobre una imagen.	70
A.6. HU Aumentar el grado de enfoque a una imagen.	70
A.7. HU Calcular la Distancia de Hamming entre dos imágenes binarizadas.	71
A.8. HU Calcular la Suma Modificada de Laplace a partir de una imagen.	71

B.1. Tarjeta CRC Detector de movimiento.	73
B.2. Tarjeta CRC Detector de volumen.	73
B.3. Tarjeta CRC Imagen de rostro.	73
C.1. Tareas de ingeniería de la primera iteración.	74
E.1. CP Detectar un rostro a partir de una imagen.	79
E.2. CP Extraer la región de los ojos.	79
E.3. CP Extraer la región de la nariz.	80
E.4. CP Extraer la región de la oreja.	80
E.5. CP Realizar el proceso de binarizado sobre una imagen.	81
E.6. CP Aumentar el grado de enfoque a una imagen.	82
E.7. CP Calcular la Distancia de Hamming entre dos imágenes binarizadas.	82
E.8. CP Calcular la Suma Modificada de Laplace a partir de una imagen.	83

Introducción

La preocupación por garantizar la seguridad en los procesos que ocurren en la sociedad moderna se ha convertido en un elemento fundamental donde las organizaciones interesadas empeñan todo el esfuerzo y recursos disponibles para alcanzar niveles aceptables en la *protección de los bienes que poseen*¹. Las *pérdidas económicas*² por violaciones de los mecanismos de seguridad han provocado que las empresas empleen métodos cada vez más sofisticados. Varios autores se refieren a *la seguridad como un proceso continuo*³ que, entre otros aspectos, define los niveles de acceso a los recursos e información en correspondencia con la importancia que la propia entidad le atribuye a cada uno; es por ello que la correcta identificación de los individuos implicados constituye un factor imprescindible para la seguridad.

Actualmente se han desarrollado diferentes métodos para la identificación de un individuo, entre los más conocidos se encuentra la creación de credenciales dentro de entornos específicos, utilizados mayormente por instalaciones que requieren un nivel de acceso limitado en algunas de sus áreas, este tipo de documento puede contener elementos de seguridad para demostrar su autenticidad y agilizar su utilización mediante sistemas de lectura mecánica o por radiofrecuencia. Sin embargo, con el desarrollo vertiginoso de las ciencias informáticas en general y la capacidad de cómputo alcanzada en los últimos años, se han desarrollado otros métodos como la biometría, cuyas potencialidades han sido incorporadas en sistemas modernos especializados en la identificación unívoca de un individuo.

La biometría es la ciencia que se dedica a medir y analizar aquellas características físicas o conductuales propias de un individuo que lo acompañan hasta su muerte. Estas características presentan un gran valor identificativo por su singularidad e invariabilidad en el tiempo. Los rasgos biométricos se pueden agrupar en *anatómicos* y de *comportamiento*, en el primer grupo se encuentran el rostro, las manos, los ojos y la huella dactilar; en el segundo grupo se incluyen la forma de caminar, la firma manuscrita y la voz. En la actualidad son muy populares los mecanismos de seguridad basados en técnicas relacionadas con la biometría, de ellos, los que se basan en el reconocimiento facial son cada vez más utilizados y más sofisticados.

Los sistemas de reconocimiento facial son aquellos capaces de identificar automáticamente a una per-

¹IBM 2015 Cost of Data Breach Study, From Ponemon Institute [3].

²Cybersecurity: The new business priority [4].

³La seguridad informática en evolución constante [5].

sona utilizando una imagen de rostro digitalizada, realizando en el proceso la extracción de aquellos rasgos faciales propios de cada persona, la fuente puede ser una imagen o un fotograma específico extraído desde un flujo de video proveniente de una cámara. El proceso de reconocimiento consiste en, inicialmente, detectar el rostro en la imagen de entrada para luego definir las características faciales de la persona a identificar. Posteriormente, se procede con la verificación de la información obtenida, se comprueba que coincidan las características almacenadas en la base de datos con las detectadas al inicio del proceso.

En la medida que las empresas implementan este mecanismo de seguridad para el control de acceso [6], los sistemas de identificación basados en imágenes faciales se han convertido en blancos de muchos incidentes de suplantación de identidad, utilizando rasgos faciales genuinos de usuarios con niveles de acceso conocidos, por lo que se evidencia la aparición de riesgos en cuanto al nivel de fiabilidad que ofrece una solución informática de este orden frente a la falsificación de los rasgos de una persona utilizando fotos o videos.

Para lograr una falsificación satisfactoria los infractores han utilizado disímiles variantes. Las fotografías son una de las formas más comunes de vulnerar los procesos de reconocimiento facial. Con el auge de las redes sociales que utilizan internet es relativamente fácil obtener imágenes de aquellas personas con determinado acceso a una organización. Empleando las herramientas adecuadas las fotos adquiridas son fáciles de editar y utilizar con fines fraudulentos; además, de forma similar a las fotos también se pueden adquirir secuencias de videos. La falsificación mediante videos permite mostrar más características físicas que cuando se utilizan las fotos, algunos elementos importantes como pestañeo de ojos, expresiones faciales y movimiento de la cabeza se pueden falsificar a través de un video cuidadosamente bien seleccionado [7].

En la actualidad es común encontrar muchos sistemas de reconocimiento facial, pero el enfrentamiento a la falsificación de características biométricas reales es muy limitado, esta característica está identificada como una cuestión medular a tener en cuenta al evaluar la calidad de un sistema de reconocimiento facial. Para combatir exitosamente los ataques por falsificación de rasgos faciales, la detección de vitalidad en el proceso de identificación es el único método de solución para verificar si la imagen a analizar fue tomada desde un rostro con vida o desde una fotografía [8]. De esta forma se puede discriminar entre aquellos rostros humanos reales y los que no lo son, dígame por utilizar fotos, máscaras

sintéticas, videos, etc., incluir este elemento dentro del procesamiento pudiera ser tenido en cuenta como un nivel de seguridad adicional al sistema de reconocimiento facial en general.

La inclusión de una técnica para la detección de vitalidad debe ser realizada al inicio del proceso de identificación. Utilizando el flujo de imágenes capturado para la verificación de la persona se calcula y evalúa el grado de vitalidad detectado, de esa forma se puede definir la validez del rostro a identificar. Basado en el resultado anterior se decide o no continuar con la verificación de la persona, evitando así realizar búsquedas innecesarias y la puesta en riesgo de la eficiencia del sistema en general.

En los sistemas biométricos de reconocimiento facial hay tres formas de implementar la detección de vitalidad en imágenes faciales; su selección está condicionada por las posibilidades o aspiraciones de los desarrolladores de software y sus clientes [8].

- **Utilizando hardware extra:** Requiere un despliegue mayor de medios técnicos, así como el incremento de los costos para su implementación. Por ejemplo algunos utilizan sistemas estéreo o cámaras de alta resolución [9].
- **Utilizando software:** Requiere menor costo en su implementación pero necesita mayor tiempo de ejecución en sus comparaciones.
- **Utilizando la combinación de técnicas:** Se puede lograr un equilibrio utilizando las características fundamentales de las anteriores [10].

El Centro de Identificación y Seguridad Digital (CISED) de la Universidad de las Ciencias Informáticas (UCI) ha desarrollado un sistema de reconocimiento de personas basado en los rasgos faciales, en el diseño inicial de este sistema no se tuvo en cuenta la inclusión de un método capaz de identificar si la imagen de entrada fue tomada de una persona viva frente a la cámara o si fue tomada presentando una foto, algún tipo de máscara o un video.

Debido a la ausencia de un mecanismo de seguridad orientado a combatir este tipo de fraude, se deriva el siguiente **problema de investigación**: ¿Cómo evitar la suplantación de identidad ante la presencia de fotos y videos en los sistemas de reconocimiento facial implementados en el CISED?

Para dar solución al problema planteado se define como **objeto de estudio** el proceso de identificación de personas mediante rasgos faciales.

El **objetivo general** de esta investigación será desarrollar un componente que permita detectar la vitalidad en las imágenes de rostro durante el proceso de captura de los sistemas de reconocimiento facial implementados en el CISED.

Para dar cumplimiento al objetivo general se han definido los siguientes objetivos específicos:

1. Confeccionar del marco teórico de la investigación para lograr un mayor entendimiento sobre el estado del arte en Cuba y el mundo referente a los sistemas de identificación utilizando rasgos faciales. Identificar los procesos principales que permiten la suplantación de identidad utilizando fotos y videos.
2. Definir las herramientas y metodologías que se utilizarán en el entorno de desarrollo.
3. Definir la arquitectura para la propuesta de solución y realizar el diseño del componente para su posterior implementación.
4. Implementar el componente que permita detectar la vitalidad en imágenes faciales, aplicando los patrones de diseño más apropiados en función de la extensibilidad de la solución y utilizar un estándar de codificación adecuado.
5. Realizar las pruebas de software al componente para verificar su correcto funcionamiento.

La detección de vitalidad en el proceso de captura de una imagen facial para la identificación de una persona utilizando fotos y videos será el **campo de acción**.

Los **métodos científicos** que se emplearán son:

Métodos Teóricos

- **Analítico:** Se utilizará en el estudio de aquellos elementos fundamentales relacionados con la detección de vitalidad, además para definir los detalles generales en cuanto a la ejecución de este tipo de análisis sobre imágenes faciales, así como los requisitos fundamentales para su ejecución.
- **Análisis Histórico - Lógico:** Se utilizará para hacer un análisis sobre los estudios existentes y el desarrollo actual que han tenido los mismos así como establecer un criterio propio sobre las tendencias que giran en torno a estos.

- **Modelación:** Se utilizará en la modelación de los diagramas y gráficas a generar para un mejor entendimiento del sistema en general.

Métodos Empíricos

- **Investigación-acción:** Para estudiar las principales vulnerabilidades del Sistema de Reconocimiento Facial del CISED referentes a la suplantación de identidad y acometer acciones en la solución presentada que aporten robustez a dicho sistema.
- **Experimental:** Se utilizará para registrar las tasas de falsa aceptación (FAR) y las tasa de falso rechazo (FRR) con o sin la utilización del componente propuesto.

Justificación de la Investigación

Los sistemas biométricos basados en rasgos faciales son de vital importancia para garantizar un nivel de seguridad alto en muchos contextos de la sociedad. De la eficiencia y el rigor con que se ejecute el proceso de verificación dependerá la calidad del resultado obtenido.

En el CISED se dispone de un Sistema de Reconocimiento Facial, dicho sistema no cuenta en su implementación con un elemento de seguridad que permita detectar ataques de suplantación de identidad utilizando la falsificación de rasgos faciales, por lo que se puede considerar muy vulnerable ante un potencial ataque de este tipo. Con el presente trabajo se pretende obtener un componente reutilizable, encargado de efectuar la detección de vitalidad utilizando una colección de imágenes consecutivas como vector de entrada, permitiendo así, detectar la presencia de imágenes faciales tomadas de individuos no vivos, de esta forma se reduce la cantidad de búsquedas realizadas sobre el sistema con el objetivo de suplantar la identidad de un usuario genuino, aportando un elemento medular en la estabilidad del sistema de identificación en general.

El tema tratado en esta investigación se encuentra en constante estudio actualmente, el impacto que supone su uso sobre los sistemas de reconocimiento facial y el valor agregado que les aporta hacen que el desarrollo e investigación con relación al tema sea tomado con la seriedad que requiere. Disponer de una solución propia, que pueda ser consultada por cualquier sistema que lo necesite y que sea capaz de solventar los problemas anteriormente aquí descritos, obliga a realizar un estudio crítico de los métodos disponibles en la bibliografía especializada, identificar las ventajas y desventajas de los

enfoques propuestos en función de los beneficios que aporta su implementación en conjunto al sistema de reconocimiento facial en general.

Estructura del documento

El presente documento se encuentra dividido en tres capítulos estructurados de la siguiente forma:

Capítulo 1: Fundamentación teórica, se describen los principales conceptos relacionados con el dominio del problema, se realiza un estudio del estado del arte sobre las técnicas para la detección de vitalidad en imágenes faciales y se analizan las principales metodologías, tecnologías y herramientas para darle solución al problema.

Capítulo 2: Características del sistema, se presentan las etapas de Planificación y Diseño descritas por la metodología XP para obtener la solución al problema planteado. Se define el modelo de dominio, las historias de usuario y los requisitos no funcionales, además se propone la arquitectura a utilizar en la implementación del componente y la planificación de las iteraciones dentro del proceso de desarrollo.

Capítulo 3: Implementación y prueba, se le da cumplimiento a la planificación mediante la implementación de la propuesta de solución y finalmente se ejecutan las pruebas de software al componente.

Capítulo 1: Fundamentación teórica

1.1. Introducción

En este capítulo se abordan los diferentes elementos teóricos que fundamentan el objeto de estudio y definen el dominio de esta investigación. Se realiza un análisis del estado del arte sobre los principales métodos utilizados a nivel global para detectar la vitalidad en los rostros humanos durante el proceso de identificación mediante imágenes faciales, se hace una valoración de las principales características de cada uno y los resultados estadísticos asociados a estos, con el objetivo de establecer una relación en busca de ventajas y desventajas que puedan conducir al cumplimiento del objetivo general de este trabajo. Finalmente se hace un estudio de las principales metodologías y herramientas necesarias para la implementación de la solución.

1.2. Algoritmos para la detección de vitalidad

La identificación facial es totalmente natural, es la menos intrusiva y más fácil de utilizar, el usuario no necesita cooperar mucho para que el proceso se ejecute completamente, basta con poder tomar una foto del rostro de la persona desde una distancia apropiada a la cámara. Desafortunadamente este método para la identificación de una persona es vulnerable ante posibles ataques utilizando la falsificación del rostro utilizando elementos como fotos, caricaturas, videos, etc. Para combatir este tipo de fraude se han propuesto métodos para detectar la falsificación de rasgos faciales, a continuación se describen algunos de los estudios más recientes.

1.2.1. Método basado en el análisis de la textura y la frecuencia

El enfoque propuesto en [11] está pensado para detectar rostros vivos y no vivos procedentes de imágenes en dos dimensiones, los autores de este método proponen un análisis del espectro formado por la imagen, explotando tanto la información de las regiones con baja frecuencia como aquellas con frecuencias más altas.

Este método se describe sobre la base de lo que se conoce como *Patrón Local Binario*(LBP)¹, el cual se emplea para analizar la textura de las imágenes con información de dos dimensiones. Por otro lado afirman que la información de la textura obtenida de imágenes tomadas de objetos de dos

¹Ver más información relacionada con el método LBP en [12].

dimensiones (especialmente aquella relacionada con la iluminación) presenta una pérdida de información con respecto a aquellas imágenes tomadas de objetos con tres dimensiones.

Para el análisis de la frecuencia se utiliza el método basado en la *Información Espectral* de dicha imagen. Los autores afirman que el análisis de la frecuencia es utilizado debido a dos causas, la primera es la existencia de aquellas características propias de los objetos con tres dimensiones, dichas características muestran una diferencia en las regiones de baja frecuencia, estando relacionadas además con la componente de iluminación observada en la forma general de un rostro real y en segundo lugar la diferencia existente a partir de la información detallada entre un rostro real y una imagen que ha sido tomada de una máscara, de esta última se pudieran observar las discrepancias en la información de alta frecuencia obtenida en cada caso [13].

Extracción de las características de frecuencia

El primer paso para ejecutar el proceso es realizar un tratamiento de la imagen obtenida, transformando el dominio de frecuencia aplicando una *Transformación Discreta de Fourier* en dos dimensiones. La imagen obtenida originalmente se puede apreciar en la Figura 1.1(a) y el resultado tras aplicar la transformación mencionada se muestra en la Figura 1.1(b). De esta forma se observa el resultado de la operación realizada sobre la imagen original, donde la frecuencia cero se encuentra en el centro de la imagen.

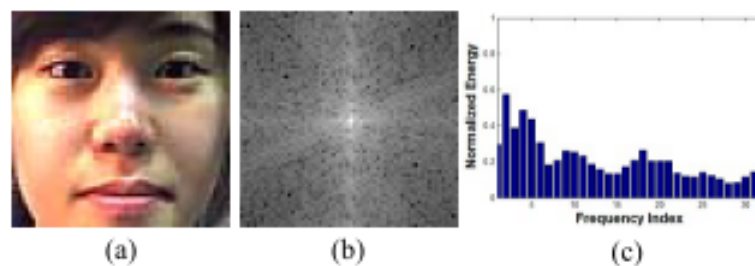


Figura 1.1: Extracción de las características de frecuencia (a)Imagen facial original (b)Imagen escalada con la Transformada de Fourier (c) 1-D vector característico de la frecuencia extraída del espectro normalizado.

El resultado de la transformación se divide en varios grupos de anillos concéntricos, donde cada anillo representa la región correspondiente a la banda de frecuencia. Finalmente, se obtiene un vector de una

dimensión combinando previamente los promedios de aquellos valores de energía de todos los anillos obtenidos. Desde luego, el promedio de valores puede variar en la medida de las diferentes regiones de frecuencia, por lo que la *normalización min-max*² puede ser utilizada.

Extracción de las características de textura

Para el análisis de las texturas se propone utilizar el método LBP siendo este uno de los más populares para describir dicho elemento de la imagen. Como se muestra a continuación en la ecuación (1.1), LBP asigna un código para cada píxel considerando la diferencia de intensidad relativa entre un píxel y sus vecinos.

$$LBR_{P,R} = \sum_{P=0}^{P-1} S(g_p - g_c)2^p, \quad s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (1.1)$$

Donde P es el número de píxeles vecinos, R es la distancia desde el centro a los píxeles vecinos. Mientras que g_c corresponde al valor de la escala de grises en el centro del píxel, g_p corresponde al valor de la escala de grises del espacio P en el anillo de radio R , entonces $S(x)$ es el resultado de la función evaluada en x . La Figura 1.2 describe la adquisición el vector característico resultado del método LBP sobre la imagen obtenida inicialmente.

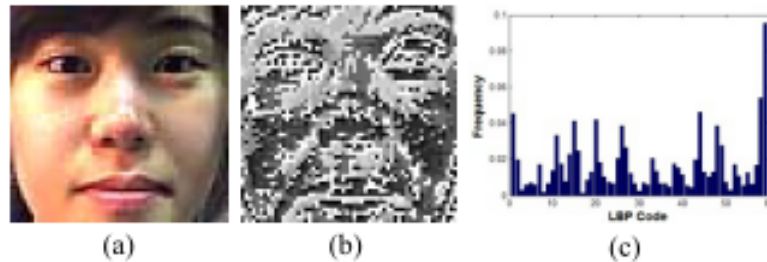


Figura 1.2: Vector característico basado en la extracción LBP (a)Imagen facial original (b)Imagen con LBP aplicado (c)Histograma del código LBP de la imagen.

² Normalización Min-Max en [14]

Fusión de los resultados de textura y frecuencia

Finalmente los valores obtenidos de ambos clasificadores serán utilizados para detectar la posible falsificación de una imagen facial[15]. Los autores de la investigación consultada, utilizaron una *Máquina de Soporte Vectorial*(SVM)³ como clasificador en el aprendizaje para la detección de vida desde una imagen en correspondencia con el vector de características basado en espectro y el método basado en LBP.

Para las pruebas se utilizaron dos tipos de bases de datos **BERC Webcam Database** y **BERC ATM Database**, cada base de datos contiene cuatro diferentes variantes de falsificación; fotos, pinturas, revistas y caricaturas. Mientras que todas las imágenes fueron capturadas en la misma resolución 640×480 , el dispositivo utilizado para cada colección fue diferente, en el primer caso se utilizó una cámara web convencional y en el otro se utilizó una cámara acoplada ATM⁴.

Pruebas y Resultados

Los resultados experimentales de la investigación mencionada arrojaron que el método basado en LBP muestra resultados más promisorios que el basado en el análisis de frecuencia cuando las imágenes son capturadas desde pinturas y caricaturas. Pero es el método basado en la fusión el que arrojó los mejores resultados con una tasa de error del 4.42% comparado con el método basado en la frecuencia con un 5.43% y el basado en LBP con una tasa de error del 12.46%.

1.2.2. Método basado en el análisis del movimiento de los ojos

Esta variante fue desarrollada por Hyung-Keun Jee[18] para un sistema integrado de reconocimiento facial. Los autores propusieron un método para detectar los ojos en una secuencia de imágenes de entrada para luego calcular la variación de cada región de los ojos y de esa forma poder determinar si la imagen tomada con la cámara fue hecha de una persona real. El fundamento para desarrollar esta alternativa lo constituye el pestañeo y los movimientos descontrolados de los ojos en los humanos, lo que genera variaciones en las figuras detectadas de cada ojo.

Pruebas y Resultados

Los resultados experimentales mostraron que cuando la puntuación se mide utilizando la distancia de Hamming, de un total de 100 rostros analizados, la puntuación media de un rostro vivo es de 30,

³<http://www.csie.ntu.edu.tw/~cjlin/libsvm> [16]

⁴Face Recognizability Evaluation for ATM Applications [17]

mientras que una rostro no vivo es 17, claramente se aprecia que la puntuación de rostros con vida es mayor a la de rostros falsos.

De acuerdo con los resultados arrojados por las pruebas realizadas durante la investigación consultada los autores afirman que cuando el umbral utilizado se fija en 21 se obtiene una tasa de FAR del 0.01 % mientras que la tasa de FRR es del 0.08 %.

1.2.3. Método basado en el análisis del enfoque variable

La técnica referida en [19] por el autor Sooyeon Kim tiene como objetivo principal aprovechar la variación de valores de píxel, utilizando dos imágenes tomadas secuencialmente con diferentes enfoques (in/out), utilizando una de las funciones de la cámara. Suponiendo que no hay una diferencia considerable en el movimiento entre ambas imágenes, los autores han tratado de encontrar la relación existente en la diferencia entre los valores de enfoque de las caras reales y las que no lo son cuando varias imágenes consecutivas son recogidas de cada sujeto.

Pruebas y Resultados

Para comprobar los resultados de las pruebas realizadas se utilizaron las tasas de falsa aceptación (FAR⁵) y falso rechazo (FRR⁶), los resultados arrojados están relacionados con el valor del *Grado de Enfoque* (DoF), por lo que según DoF se hace menor, entonces FAR es de 2.86 % y FRR es 0.00 %, en la medida que el valor DoF aumenta el promedio de FAR y FRR se incrementa. En este sentido se deduce que los resultados experimentales muestran que este método es dependiente del valor DoF, en la medida que se quieran obtener mejores resultados es vital disminuir el valor de DoF utilizado.

1.2.4. Método basado en el pestañeo de los ojos

El empleo de este método se basa en un reflejo incondicional de los ojos que ayuda a limpiar y eliminar partículas irritantes de la superficie de la córnea y la conjuntiva, la velocidad con que se ejecuta esta acción involuntaria puede variar según el estado físico de la persona, pudieran influir elementos como la fatiga, el estrés emocional, acumulación de sueño, algún tipo de medicamento específico, alguna

⁵Acrónimo de False Accepted Rate, es el error que se produce cuando el sistema indica que la información adquirida del usuario en la entrada sí se corresponde con la plantilla almacenada, cuando realmente se trata de otra persona

⁶Acrónimo de False Reject Rate, es el error que se produce si el sistema indica que el usuario en la entrada no se corresponde con la plantilla almacenada, y realmente sí es la misma persona

enfermedad, etc. en las investigaciones consultadas [20, 21, 22, 23] se afirma que una persona pestañea de forma incondicional o no un promedio de 15 o 30 veces en el espacio de 1 minuto. Esto quiere decir que una persona cierra y abre los ojos por lo menos una vez en el intervalo de 2 a 4 segundos y el tiempo que se emplea en cada pestañeo está cerca de los 250 milisegundos. Con una cámara web de uso común es posible obtener 15 cuadros por segundo, donde el intervalo entre cuadros no supera los 70 milisegundos, por lo que resulta sencillo con un dispositivo de este tipo capturar 2 o más cuadros por cada pestañeo. Basados en lo anterior los autores afirman que se hace factible utilizar este método para detectar imágenes de rostros no vivos.

Los autores de la investigación mencionada hicieron una comparación del método *Conditional Random Fields*(CRFs)⁷ con un modelo discriminativo como el *AdaBoost* y un modelo generalizador como el *Hidden Markov Model*(HMM)⁸.

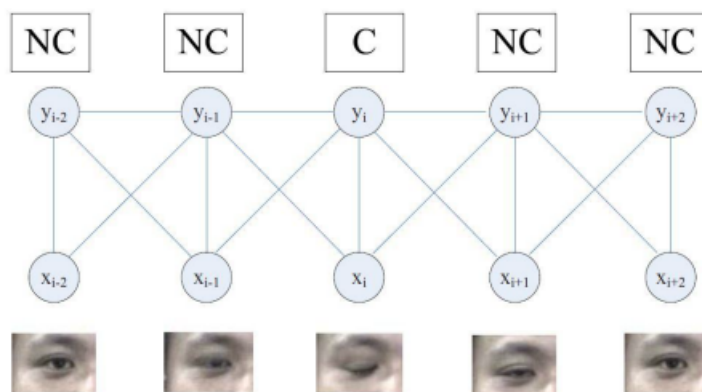


Figura 1.3: Estructura gráfica del pestañeo de ojos basado en el modelo CRF. Se utiliza la notación C y NC para ojos cerrados y no cerrados respectivamente.

En la investigación se empleó una estructura lineal de CRF, de acuerdo con los autores el estado donde los ojos se encuentran medio abierto es muy difícil de detectar en diferentes individuos, debido a que el tamaño de los ojos depende de la apariencia de cada persona, por ejemplo el estado abierto en un ojo pequeño pudiera ser similar al estado medio abierto de un ojo de mayor tamaño. En la Figura 1.3 se considera $Y = (y_1, y_2, \dots, y_T)$ para la secuencia de etiquetas de estados, $X = (x_1, x_2, \dots, x_T)$ será la

⁷Probabilistic Models for Segmenting and Labeling Sequence Data [24]

⁸Hidden Markov Models [25]

secuencia de observaciones.

Pruebas y Resultados

Para las pruebas se utilizó una base de datos⁹ que se encuentra disponible en línea. La misma tiene un total de 80 videos de unos 20 individuos diferentes y 4 videos por cada individuo. Cada video tiene una velocidad de 30 marcos por segundo durante 5 segundos con una dimensión de 320×240 . El número de pestaños en un vídeo puede variar de 1 a 6 veces, para un total de 256 pestaños en la base de datos, toda la información fue recolectada sin condiciones especiales de iluminación.

Se hicieron comparaciones del enfoque CRF con respecto al método Adaboost y HMM. Los promedios de detección obtenidos en cada prueba se muestran en las Tablas (1.1), (1.2). Como conclusión los autores afirman que este enfoque tiene algunas limitaciones con respecto al uso de gafas con alto grado de reflexión, las que puedan cubrir total o parcialmente la región de los ojos.

Estilos Diferentes	Cascade AdaBoost	HMM	CRF(w=2)
Sin gafas	100 %	100 %	100 %
Con gafas finas	95 %	95 %	90 %
Con gafas más oscuras	80 %	85 %	100 %
Sin gafas y mirando hacia arriba	85 %	95 %	100 %

Tabla 1.1: Detección rostro vivo para los métodos comparados.

Cascade AdaBoost	HMM	CRF(w=2)
95 %	87,8 %	98,3 %

Tabla 1.2: Detección de fraude para los métodos comparados.

1.2.5. Método basado en la clasificación binaria

Esta técnica fue tratada por Xiaoyang Tan en [1], basados en dos criterios (1) un rostro real es un objeto 3D, mientras que los rostros no vivos es decir aquellos detectados desde una foto tomada con anterioridad son objetos 2D; (2) la aspereza de las superficies son diferentes en ambos casos. En esta propuesta de solución también se adopta un método de análisis en tiempo real, no intrusivo y con poca colaboración del usuario utilizando una cámara web convencional. El proceso ha sido definido como

⁹ZJU eyeblink database en [26]

un problema de clasificación binaria y utilizando el modelo de Lambert, se proponen dos estrategias para extraer la información fundamental sobre las propiedades de las diferentes superficies de un rostro genuino y una fotografía.

Para evaluar este método los autores construyeron una base de datos de impostores la que se encuentra disponible en línea¹⁰ y la misma se realizó utilizando una cámara convencional adquirida en una tienda de productos electrónicos. La información de esta base de datos fue recolectada en tres sesiones con dos semanas entre dos sesiones, donde el lugar y los niveles de iluminación fueron diferentes. Se usaron quince sujetos y en cada sesión se tomaron fotos en vivo de cada sujeto y también de fotos suyas. Algunos estudios preliminares arrojaron que este enfoque muestra buen comportamiento, con ventajas como la poca intrusión y la no necesidad de hardware extra en su implementación. Los resultados de la prueba realizada se pueden ver en la Tabla (1.3).

Aunque los autores obtuvieron buenos resultados en las pruebas realizadas pasaron por alto el problema de las condiciones de mala iluminación, en otra investigación Peixoto[2], se extendió este enfoque para ser utilizado en entornos con niveles bajos de iluminación. La idea principal de este último estudio se basa en que el brillo de la imagen capturada desde una pantalla LCD afecta la imagen en su estructura, de una forma en que las regiones de alta frecuencia son propensas a un efecto de emborronamiento debido a los píxeles con niveles altos de brillo con respecto a su vecino. Esto provoca que las imágenes falsas muestren menos bordes que las imágenes de rostros reales.

Basados en esta información los autores pudieron detectar si la imagen de entrada proviene de un engaño y cuando no. Primeramente analizaron la imagen utilizando un filtro de *Diferencia Gaussiana* (DoG) que emplea dos filtros con diferente desviación estándar como límite. La idea principal fue mantener los niveles de frecuencia altos y medios con el objetivo de eliminar el ruido, pero en este caso (DoG) no detecta los bordes apropiadamente cuando se encuentra bajo condiciones de mala iluminación. Para la clasificación el autor utiliza el modelo de regresión logística esparcida, similar a como se hizo en la investigación [1]. Posteriormente con el objetivo de minimizar el problema de la iluminación defectuosa la imagen fue procesada con la idea de hacerla homogénea, para que los cambios de iluminación estén más controlados. Se utilizó un límite de contraste adaptativo en la ecualización del histograma cuyo

¹⁰<http://parnec.nuaa.edu.cn/xtan/data/NUAAImposterDB.html>

propósito general es operar en las regiones pequeñas de la imagen, llamadas *Tiles*. Los resultados experimentales utilizando esta extensión se pueden ver en la Tabla (1.4).

	Mínimo	Medio	Máximo	STD
Precisión de la clasificación	85,2 %	86,6 %	87,5 %	0,6 %
Números de Verdaderos Positivos	81,9 %	82,4 %	90,4 %	0,6 %
Números de Falsos Positivos	8,0 %	9,3	18,8 %	1,3 %

Tabla 1.3: Comportamiento de [1] utilizando NUAA Imposter Database.

	Mínimo	Medio	Máximo	STD
Precisión de la clasificación	92,0 %	93,2 %	94,5 %	0,4 %
Números de Verdaderos Positivos	92,6 %	93,0 %	93,7 %	0,4 %
Números de Falsos Positivos	4,7 %	6,7	8,4 %	1,3 %

Tabla 1.4: Comportamiento de [1] aplicando la extensión sugerida por Peixoto en [2].

1.2.6. Método basado en el componente dependiente

En la investigación [27] se plantea un enfoque donde se definen cuatro pasos, (1) localizar los componentes del rostro, (2) codificar las características a bajo nivel, (3) derivar la representación del rostro a alto nivel poniendo en común los códigos con las ponderaciones derivadas del criterio de Fisher y (4) la concatenación de los histogramas de todos los componentes en un clasificador para la identificación.

Los autores detectaron que es significativa la diferencia operacional entre los rostros genuinos y los que no lo son, los primeros son capturados por la cámara de una vez, mientras que los segundos se obtienen por la re-captura de imágenes utilizando fotos o videos en pantallas. Según su investigación afirman que se producirán diferencias en tres aspectos, (1) las caras son borrosas a causa de la resolución limitada de las fotos o pantallas y el re-desenfoco de la cámara, (2) la apariencia de los rostros pudieran variar más o menos en función del cambio de reflectancia causada por la corrección gamma de la cámara, (3) la apariencia también pudiera verse afectada por el sombreado anormal existente en la superficie de las fotos y pantallas. Finalmente se utiliza una máquina de soporte vectorial como clasificador. El proceso completo para la clasificación de la imagen de entrada se describe en la Figura 1.4.

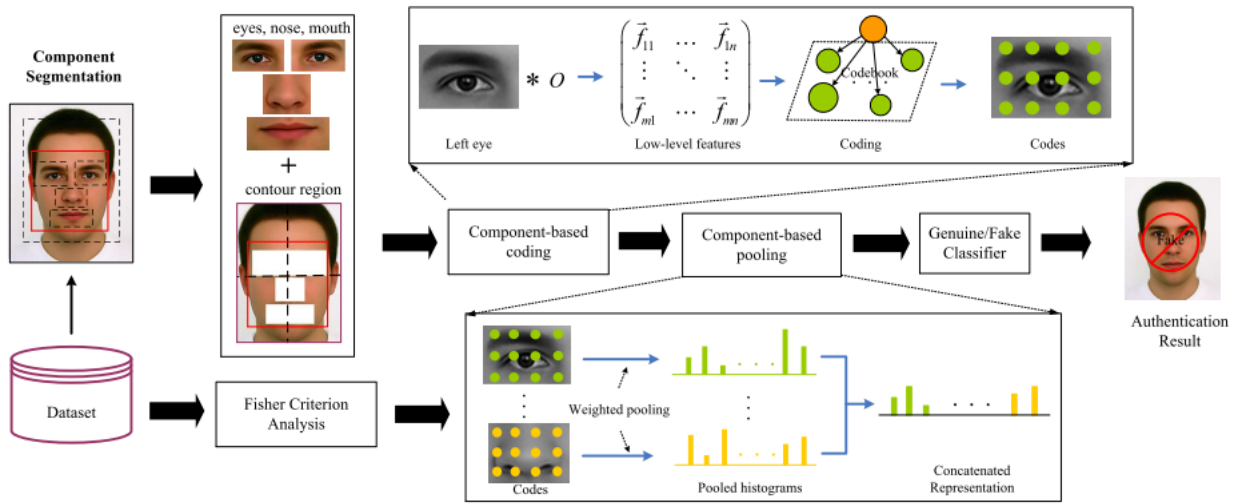


Figura 1.4: El flujo del análisis realizado en el enfoque propuesto, O representa un operador de características.

Pruebas y Resultados

Para los experimentos los autores utilizaron tres tipos de bases de datos, NUAA Database, CASIA Database y Print-Attack Database. En el caso de las pruebas utilizando NUAA Database, la habilidad del discriminante en imágenes aplicándose H-Faces es mejor con respecto a imágenes con las dimensiones originales utilizando características basadas en el método de Diferencia Gaussiana *DoG baseline*[28] y características basadas en el método *MsLBP*[29], en la Tabla (1.5) se aprecia como el enfoque propuesto utilizando el criterio de *weighted pooling* obtuvo mejores resultados.

Métricas	DoG, Face	DoG, H-Face	MsLBP	MsLBP, H-Face	Método Propuesto
Precisión	0,746	0,818	0,749	0,927	0,977
Área Bajo la Curva (AUC)	0,717	0,830	0,873	0,990	0,998
Tasa de Error (EER)	0,359	0,233	0,239	0,048	0,019

Tabla 1.5: Comportamiento al ejecutar las pruebas utilizando NUAA Database.

Para las pruebas utilizando *PRINT-ATTACK Database* se siguió el protocolo recomendado, se programaron tres experimentos con diferentes selecciones de rostros, (1) G+F, "fixed" sub-database, (2) G+H, "hand" sub-database y (3) G+F+H, la unión de ambos. Los resultados también fueron comparados con los métodos DoG y MsLBP como se ve en la Tabla (1.6) el método propuesto arrojó mejores resultados

en los tres experimentos.

Escenario	Métricas	DoG, Face	DoG, H-Face	MsLBP	MsLBP, H-Face	Método Propuesto
G+F	Precisión	0,819	0,844	0,918	0,971	0,995
	Área Bajo la Curva (AUC)	0,978	0,990	0,980	0,988	1,000
	Tasa de Error (EER)	0,068	0,055	0,068	0,026	0,003
G+H	Precisión	0,852	0,884	0,898	0,958	0,991
	Área Bajo la Curva (AUC)	0,957	0,977	0,987	0,996	1,000
	Tasa de Error (EER)	0,127	0,097	0,066	0,035	0,009
G+F+H	Precisión	0,847	0,900	0,885	0,962	0,988
	Área Bajo la Curva (AUC)	0,967	0,983	0,991	0,994	0,999
	Tasa de Error (EER)	0,096	0,073	0,049	0,038	0,012

Tabla 1.6: Comportamiento al ejecutar las pruebas utilizando PRINT-ATTACK Database.

Los resultados utilizando *CASIA Database* se determinaron utilizando un protocolo parecido al anterior, en este caso con seis escenarios. Se trabajó con la *Calidad Baja*(LQ), *Calidad Moderada*(MQ), *Calidad Alta*(HQ), *Foto Distorsionada*(WP), *Foto Recortada*(CP), *Foto Video*(VP) y la combinación de todos como se muestra en la Tabla (1.7).

Escenario	Métricas	DoG, Face	DoG, H-Face	MsLBP	MsLBP, H-Face	Método Propuesto
Todos	Precisión	0,760	0,802	0,823	0,851	0,898
	Área Bajo la Curva (AUC)	0,754	0,801	0,884	0,915	0,941
	Tasa de Error (EER)	0,306	0,267	0,185	0,157	0,118

Tabla 1.7: Comportamiento al ejecutar las pruebas utilizando CASIA Database.

Como se puede observar los autores validaron su propuesta con tres bases de datos diferentes, en el enfoque de su investigación introdujeron la utilización de los H-Face con lo que se obtuvo mayor información de cada imagen de entrada, además se utilizó el criterio de Fisher para dirigir el procesamiento, arrojando el método propuesto el mejor comportamiento en cada una de las comparaciones realizadas con los otros enfoques.

1.2.7. Método basado en el movimiento de los labios.

Este enfoque fue tratado por Kollreider[30], aprovechando el movimiento de los labios. Para la clasificación los autores sugieren utilizar una SVM¹¹, específicamente proponen localizar la región de la boca

¹¹Máquina de Soporte Vectorial

y extraer el *flujo óptico de líneas* (OFL) en tiempo real. Para las pruebas a este mecanismo de detección varias personas fueron grabadas pronunciando los dígitos del 0 al 9, con el objetivo de detectar el nombre de cada dígito en el movimiento de los labios. Después de un análisis de los valores detectados y la creación de 10 clases de SVM, se obtuvieron las matrices de confusión, los resultados cuantitativos de las pruebas en base a 100 individuos, están sobre el 73 % del índice de identificación, es por esto que los autores lo proponen como un método para la detección de fraudes.

1.2.8. Análisis basado en la combinación de técnicas.

Según Kollreider[31] donde se aplicó la combinación de varias técnicas que fueron probadas en tiempo y escenarios reales para detectar posible fraudes de identificación en espacios interiores, plantean que primeramente era necesario que el algoritmo utilizado fuera capaz de detectar un rostro en la imagen obtenida de la cámara, luego se inicia el proceso de detección de evidencias que indiquen que es una persona viva. Posteriormente sería analizada la posibilidad de que sea la foto de una persona con vida, comprobando así la existencia de elementos que indiquen volumen, pestañeo de ojos o movimiento de la boca, siempre de la forma menos intrusiva sobre el cliente. Si no se encuentra dicha respuesta entonces las respuestas se piden y se comprueban de forma aleatoria. Después de expirado el tiempo se verifica la vitalidad del rostro detectado. Para la experimentación, se empleó una webcam de bajo costo que obtiene marcos de 320×240 píxeles a 25 marcos por segundo y la capacidad de cómputo empleada se basa en un ordenador portátil estándar. Los autores sugirieron que el rendimiento mostrado por el método propuesto es eficiente.

1.3. Análisis de las investigaciones consultadas.

Durante la presente investigación se han señalado aquellas características de los enfoques para la detección de vitalidad en imágenes faciales con resultados estadísticos más promisorios dentro de la bibliografía consultada.

Dentro de las investigaciones analizadas se destacan tres indicadores fundamentales, ver Tabla (1.8), *los movimientos, las texturas y los signos de vida*. El análisis del movimiento depende generalmente del flujo óptico calculado a partir de secuencias de vídeo. Cuando se realiza el análisis del movimiento, es muy difícil hacer una suplantación utilizando una imagen de un rostro en 2D, esta revisión es independiente de la textura y no es necesaria la colaboración del usuario. En el estudio del movimiento se necesita

efectuar un análisis de la secuencia de video y es difícil hacerlo cuando el flujo de imágenes tiene baja actividad de movimiento. Este enfoque puede ser vulnerado por esculturas 3D y necesita imágenes de alta calidad.

Indicador de vida	Ventajas	Desventajas
Análisis de texturas	Fácil de implementar y no necesita la colaboración del usuario.	Difícil de utilizar con imágenes con poca información de textura y la diversidad en las imágenes capturadas.
Análisis de movimientos	Independiente a las texturas, difícil de falsificar usando imágenes 2D y no necesita la colaboración del usuario.	Necesita un flujo de video y es vulnerable a falsificaciones con elementos que contengan dimensiones 3D.
Análisis de signos de vida	Difícil de falsificar utilizando figuras en 2D y 3D e independiente a las texturas.	Se necesita la colaboración del usuario, depende de la detección de segmentos de rostro y necesita una secuencia de video.

Tabla 1.8: Ventajas y desventajas de los enfoques para la detección de vitalidad.

Por su parte las técnicas enfocadas en el análisis de textura aprovechan los patrones perceptibles en la imagen. Este tipo de análisis se basa en el criterio de que las fotos de rostros humanos tomadas de objetos planos son significativamente diferentes con respecto a aquellas que son tomadas de objetos tridimensionales. Es útil emplear este enfoque para identificar caras reales dado que las impresiones en papel generalmente contienen características de texturas altas. Este método es relativamente fácil de implementar y no necesita la colaboración del usuario para efectuar la recolección de evidencias. Aunque es válido destacar la existencia de una amplia diversidad de texturas debido a los diferentes tipos de impresión, por lo que estos sistemas deben ser lo suficientemente robustos como para enfrentarse a tan diverso número de patrones. La otra posibilidad es que un ataque pudiera ser realizado empleando la pantalla de un dispositivo donde la información de texturas es muy limitada.

En el caso de la detección de vida podemos encontrar dos formas de tratar el problema. La primera forma es asumiendo algún movimiento conocido de las personas. En este caso, al usuario se le indica

realizar una tarea determinada para verificar la viveza de la foto de su rostro, esta tarea puede ser algún tipo de movimiento que se considera como una respuesta al desafío impuesto. En el caso de los usuarios que logren completar la tarea se clasificaría su rostro como verdadero. En el otro enfoque asociado a este método no se asume ninguna característica o interacción del usuario como regla, pero se centra en gestos involuntarios de ciertas partes de la cara como el movimiento de los ojos, la boca o el parpadeo y serán considerados signos para ser clasificado como una imagen proveniente de un rostro con vida. Este enfoque resulta más difícil de vulnerar utilizando métodos más sofisticados como imágenes en 2D y esculturas o máscaras en 3D.

1.4. Metodología de desarrollo de software.

Una metodología de desarrollo de software es un conjunto de pautas o procedimientos previamente descritos con la pretensión de regir, planificar y controlar el proceso de desarrollo. El objetivo principal es guiar a los ingenieros de software en la obtención de un producto con la calidad requerida, aprovechando al máximo los recursos asignados y que cumpla con los requisitos del cliente. Cada sistema a construir tiene características propias y requiere tecnologías afines en cada caso, debido a esto se han diseñado diferentes metodologías con el objetivo de adecuarse mejor a cada problema en específico, actualmente existen dos grupos, las metodologías tradicionales o pesadas y las metodologías ágiles o ligeras [32].

1.4.1. Metodologías ágiles.

Las metodologías ágiles surgen como resultado de la amplia gama de opiniones sobre las ventajas y desventajas que proporciona el uso de una metodología tradicional, enfocándose en dos aspectos importantes, el retrasar las decisiones y la planificación adaptativa [32].

Como resultado del nuevo enfoque surge un nuevo *Manifiesto Ágil*¹² cuyas principales ideas son:

- Los individuos y las interacciones entre ellos son más importantes que las herramientas y los procesos empleados.
- Es más importante crear un producto software que funcione que escribir documentación exhaustiva.
- La colaboración con el cliente debe prevalecer sobre la negociación de contratos.

¹²The Agile Manifesto [33].

- La capacidad de respuesta ante un cambio es más importante que el seguimiento estricto de un plan.

La utilización de una metodología ágil es recomendable cuando se trabaja en entornos donde los requisitos pueden cambiar eventualmente, debido a que la prioridad de este tipo de enfoque es la respuesta rápida y flexible en lugar de seguir estrictamente la planificación previa, además resulta muy efectivo cuando se trabaja en equipos de desarrollo relativamente pequeños.

Extreme Programming (XP)

Esta metodología es la más destacada entre los enfoques ágiles, está centrada en facilitar las relaciones entre los trabajadores, promoviendo el trabajo en equipo y propiciando un entorno de trabajo agradable. La programación extrema asigna mayor importancia a la adaptabilidad que a la previsibilidad. Los autores que favorecen a XP consideran que los cambios en los requisitos de forma improvisada son un proceso natural en el proceso de desarrollo y creen tener la facilidad de adaptarse en cualquier momento del ciclo de vida del proyecto. XP presenta según se muestra en la Figura 1.5 un ciclo de desarrollo iterativo incremental, integrado por seis fases: Exploración, Planificación de la entrega, Iteraciones, Producciones, Mantenimiento y Muerte del proyecto.

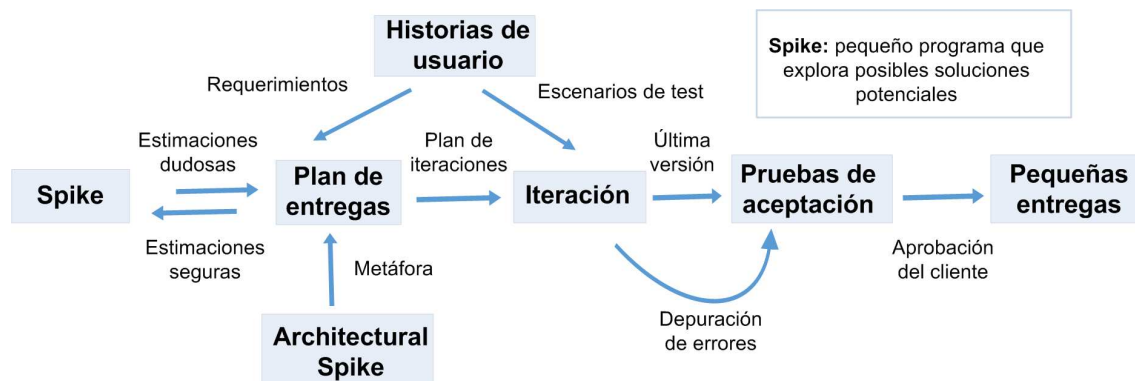


Figura 1.5: Flujo de trabajo en XP.

Este enfoque promueve una serie de prácticas entre las que se encuentra la codificación en parejas lo que permite la discusión del código generado mientras se escribe, además requiere pequeñas entregas a los clientes de versiones operativas aunque no incluyan las funcionalidades demandadas en su totalidad.

Aboga por un diseño simple y código refactorizado con el objetivo de facilitar posibles cambios en el futuro, por otro lado, la presencia de un representante del cliente dentro del grupo de desarrollo es vital para la discusión de posibles cambios de forma efectiva, así como también la inclusión de estándares de codificación para asegurar la compresión del código escrito [34].

A lo largo del ciclo de desarrollo es de vital importancia la retroalimentación entre el cliente y el equipo de desarrollo, esto asegura que el cliente guíe el trabajo hacia aquellos aspectos de mayor incidencia en el negocio, tiene además la potestad de velar por sus intereses y que aquellas reglas de su negocio sean debidamente incluidas en la solución, mientras que el equipo de desarrollo tiene el rol de estimar el esfuerzo necesario en la implementación. La clave del éxito del enfoque propuesto por XP está en la buena comunicación entre las partes implicadas, ese es el elemento que asegura la respuesta eficiente a los cambios de último momento [35].

Durante la implementación del componente que se propone como resultado de la investigación, se determinó el uso de la metodología XP en función de las ventajas que ofrece, específicamente cuando se tiene un equipo de desarrollo con una cantidad de miembros limitada, también resulta importante destacar que la documentación a generar en este caso será la mínima necesaria para comprender el proceso de desarrollo. Además existe en el CISED una experiencia previa relacionada con el uso de la metodología seleccionada, lo que facilita el trabajo e impone al equipo de desarrollo una curva de aprendizaje relativamente baja, esta última condición evita que el equipo de trabajo tenga que invertir tiempo extra en comprender sus lineamientos.

1.5. Herramientas y tecnologías para la solución del problema.

Las tecnologías y herramientas son fundamentales en los procesos de desarrollo de software, en correspondencia con la selección que se haga, de la forma en que se utilicen y el dominio que tengan los miembros del equipo de trabajo dependerá el tiempo total que se invertirá para obtener la solución, además permitirá explotar sus potencialidades al máximo evitando programar tareas de forma innecesaria, reflejándose en la calidad del producto final.

1.5.1. Lenguaje de modelado.

Utilizar un lenguaje de modelado durante el proceso de diseño e implementación del producto es muy importante. Se trata de un tipo de notación que incluye símbolos y formas, capaces de describir,

organizar y estructurar lógicamente el comportamiento de los componentes en la solución.

Unified Modeling Language (UML)

UML es un lenguaje de modelado que incluye y estandariza todas aquellas técnicas que han sido utilizadas hasta el momento para el modelado de sistemas y procesos. Permite construir y documentar un software especificando cada una de sus partes. Este lenguaje contiene todos aquellos conceptos necesarios para describir detalladamente los procesos del negocio, funcionalidades concretas de la aplicación, expresiones del lenguaje de programación, plataformas de desarrollo, etc. UML en si mismo no constituye un lenguaje de programación pero si un potente lenguaje para el modelado de propósito general, demostrando ser una herramienta muy efectiva en los procesos de ingeniería de software [36].

1.5.2. Herramienta CASE.

Las herramientas para realizar Ingeniería de Software Asistida por Computadoras (CASE) por sus siglas en inglés, disponen de las técnicas, métodos y utilidades orientadas a la automatización del ciclo de vida un software. Haciendo uso de este recurso se puede reducir el tiempo invertido en el proceso de desarrollo, por lo que los niveles de productividad se ven beneficiados [37].

Visual Paradigm

Esta herramienta pertenece a Visual Paradigm International, soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientado a objetos, construcción, pruebas y despliegue. Soporta un grupo significativo de diagramas y otros artefactos, permite hacer ingeniería directa e inversa y genera además la documentación relacionada en el proceso de ingeniería en varios formatos, además el equipo de desarrollo dispone de experiencia previa positiva utilizándola.

1.5.3. Lenguaje de programación.

Los lenguajes de programación son intermediarios entre el programador y el ordenador, utilizándolos se pueden crear programas de computadoras que regulen o controlen el comportamiento del hardware y el software. Los lenguajes tienen estructuras gramaticales, símbolos, reglas semánticas y expresiones propias que les caracterizan, las que permiten además leer, interpretar y mantener el software.

CSharp (C#)

Este lenguaje es desarrollado por la compañía *Microsoft Corporation*¹³ para ser utilizado en su plataforma *.Net*¹⁴ y responde al paradigma orientado a objetos. Desarrollar utilizando este lenguaje es relativamente sencillo, su sintaxis está basada en otros lenguajes más antiguos como C++ y Java, aunque inicialmente solo estaba disponible sobre el Sistema Operativo Windows, en la actualidad ya existe un compilador para su uso sobre las plataformas Unix, MacOS y la familia de GNU/Linux [38].

1.5.4. Entorno de desarrollo integrado (IDE).

Un entorno de desarrollo integrado o IDE (Integrated Development Environment), por sus siglas en inglés, es un software que incluye las herramientas necesarias para que los programadores puedan codificar de forma cómoda y organizada. Disponen de un editor de texto especial que permite la representación gráfica de las características propias del lenguaje utilizado, además poseen todas las herramientas para interpretar, compilar o depurar el código escrito así como un asistente para crear las interfaces gráficas, en algunos casos se les ha incluido compatibilidad con los diferentes sistemas de control de versiones.

Microsoft Visual Studio

Este IDE es desarrollado por la compañía norteamericana Microsoft, está orientado a satisfacer las necesidades de alta disponibilidad, rendimiento, escalabilidad y complejidad. Incluye todas las herramientas necesarias para elevar los niveles de productividad y desarrollar aplicaciones con calidad. Se puede utilizar para implementar tanto aplicaciones locales como sitios y aplicaciones web. Soporta el trabajo en lenguajes de programación como C++, C#, Visual Basic .Net, F#, Java, Python, Ruby, PHP; al igual que entornos de desarrollo web como ASP.Net MVC, Django, etc [39, 40].

Aprovechando las potencialidades que ofrece como editor de código y gestión del entorno de trabajo, permitirá establecer un ciclo de desarrollo ágil, obteniendo un producto eficiente y confiable teniendo en cuenta el lenguaje de desarrollo seleccionado, además valorando que la propuesta de solución debe ser utilizada como un componente del Sistema de Reconocimiento Facial del CISED y este último funciona sobre la plataforma .Net de Microsoft, el factor de compatibilidad deseado en la posterior integración

¹³<http://www.microsoft.com/en-us/default.aspx>

¹⁴<http://www.microsoft.com/net>

puede ser mantenido en todo momento del ciclo de desarrollo.

1.5.5. Procesamiento digital de imágenes.

Es el conjunto de técnicas que se aplican a las imágenes digitales con el objetivo de mejorar su calidad o hacer más evidentes en ellas ciertos detalles que se desean analizar o resaltar. El procesamiento digital de imágenes se efectúa dividiendo la imagen en un arreglo rectangular de elementos y cada elemento se denomina píxel.

Open Source Computer Vision Library (OpenCV)

Es una librería inicialmente desarrollada por la compañía *Intel Corporation*, en la actualidad está liberada con *licencia BSD*¹⁵ por lo que su uso es gratis tanto para las soluciones informáticas de orden académico como para aquellas con fines comerciales, está escrita en lenguaje C y C++. Está diseñada para alcanzar la mayor eficiencia computacional y mantiene un enfoque orientado a las aplicaciones en tiempo real. Tiene interfaces para interactuar con lenguajes como C++, C, Python y Java, además es compatible con Microsoft Windows, Linux, MacOS, iOS y Android. Entre las principales utilidades de las que dispone se encuentran las herramientas para el trabajo con funciones en 2D y 3D, uso en sistemas de reconocimiento facial, reconocimiento de gestos, seguimiento de movimientos, etc [41].

Se ha seleccionado esta librería teniendo en cuenta que dispone de una amplia gama de funciones, siendo estas las herramientas necesarias para la aplicación de las técnicas de detección de características en las imágenes de rostros. Esta librería posee una interfaz para el trabajo con el lenguaje de programación seleccionado, estando escrita en C y C++ su rendimiento es eficiente frente los análisis en tiempo real que requiere la solución propuesta.

1.6. Conclusiones parciales.

Se ha realizado un análisis de los diversos enfoques orientados a la detección de vitalidad en imágenes faciales, se utilizó como elemento de referencia en cada investigación previa consultada el tipo de técnica utilizada y los tipos de indicadores o pistas empleados para identificar la falta de vida en un rostro, así como la relación de cada técnica con los diferentes tipos de falsificaciones y sus peculiaridades en las soluciones desarrolladas; en esta revisión se evaluaron aquellos trabajos más interesantes y prometedores para dar respuesta a la problemática planteada en el diseño teórico de esta investigación.

¹⁵<http://opencv.org/license.html>

En el transcurso de este análisis se han detectado algunos elementos comunes que afectan el rendimiento de varias de las técnicas aquí descritas, tales son:

- Los efectos del cambio de iluminación.
- Los efectos del ruido amplificado en las imágenes.

Estos elementos alteran la información de la textura extraída en la imagen de entrada. En el caso del método de detección que utiliza el movimiento de los ojos, si el cliente porta elementos como gafas o espejuelos estos pudieran provocar algunos efectos de reflexión en la imagen. Los autores en las investigaciones relacionadas con este tipo de enfoque sugieren que este fenómeno sea debidamente analizado y tratado en las posteriores investigaciones referentes a la temática que nos ocupa. Por otra parte, los conjuntos de datos utilizados para entrenar y preparar los sistemas de detección de vitalidad necesitan ser lo suficientemente diversos e informativos como para imitar los escenarios reales.

En esta fase de la investigación se analizaron las tecnologías, lenguajes, metodologías y otras herramientas a emplear, teniendo en cuenta además las políticas de desarrollo del CISED y lo conveniente de su selección se decidió utilizar XP como la metodología de desarrollo, Visual Paradigm como la herramienta CASE para el modelado de la solución propuesta, C# como lenguaje de programación incluyendo la librería OpenCV y Visual Studio como entorno de desarrollo integrado.

Capítulo 2: Características del sistema.

2.1. Introducción

En este capítulo se tratarán las características fundamentales de la propuesta de solución. Serán descritos los principales conceptos del modelo de dominio, se especifican los requisitos funcionales y no funcionales del componente y serán representados en la historia de usuario que le corresponde. Se elaboran los diagramas de clases del diseño teniendo como referencia las tarjetas CRC, además se hace una descripción de la arquitectura propuesta. Se realiza también un análisis de los patrones de diseño utilizados y su desempeño en el funcionamiento de la solución.

2.2. Propuesta de solución

En la propuesta de solución se incluirán dos métodos seleccionados a partir del estudio realizado, en el que fueron valoradas las ventajas y desventajas que ofrece cada uno y el nivel de complejidad que poseen en función del tiempo previsto para esta investigación, se tuvo en cuenta además la cantidad de miembros del equipo de desarrollo. Los resultados de la investigación estarán orientados específicamente a la detección de fraudes utilizando fotos y videos. Para la detección de vitalidad ante fotos se decidió utilizar el estudio del movimiento en la región de los ojos, mientras que para la detección de fraudes utilizando videos, se decidió incluir en la solución propuesta el análisis de la variación de enfoque en regiones diferentes del rostro.

2.2.1. Método basado en el análisis del movimiento de los ojos

Del autor Hyung-Keun Jee[18] que utiliza como enfoque el análisis de texturas, analizado en la sección 1.2.2 y que permite de forma poco intrusiva detectar la existencia de cambios en la región de los ojos, el autor afirma que realizando el análisis descrito en su método, dichos cambios pueden ser interpretados como señales de vida.

Procedimiento

El primer paso es detectar el punto central de cada ojo en la imagen de entrada utilizando un mecanismo para la detección de ojos, se normaliza dicha región y se extrae aquella subregión que contiene la información de cada ojo. Luego de binarizar cada una de las regiones extraídas como se muestra en la Figura 2.1, son comparadas calculando la variación existente entre aquellas que representen el mismo

ojo. En caso que el resultado de la variación obtenida tras las comparaciones sea mayor que un umbral se puede clasificar la imagen de entrada como válida según el flujo descrito en la Figura 2.2, en caso contrario se puede decir que fue tomada desde una fotografía o un rostro no vivo.

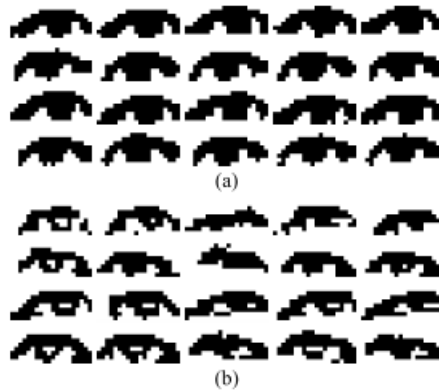


Figura 2.1: Binarización de imágenes falsas en (a) y en imágenes reales en (b).

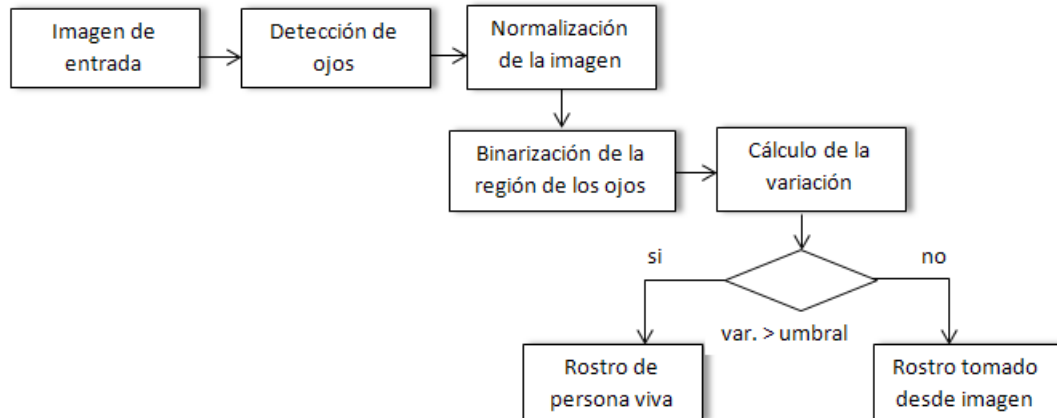


Figura 2.2: Flujo para la clasificación de las imágenes.

Características e implementación

Para la detección de las regiones de los ojos, los autores utilizaron el hecho de que la intensidad de la región de cada ojo es más baja que el resto de la región de la cara si la imagen es considerada una curva 3D, para realizar la detección de los ojos primeramente se aplicó un filtro gaussiano a la imagen, así

se obtiene la curva 3D necesaria. En dicha curva es necesario extraer todos aquellos mínimos locales utilizando el método del gradiente descendente[42], para reducir la cantidad de regiones no válidas clasificadas como posibles ojos candidatos, el clasificador de ojos fue preparado con el método de entrenamiento *Vilova's AdaBoost*[43]. Para atenuar el efecto de la iluminación se le aplicó a la imagen de entrada un *Self Quotient Image (SQI)* el que pudiera ser pensado como un filtro de paso alto[44]. El umbral es adaptativo y se obtiene del valor medio de píxel de cada región de ojo tomada tras ser binarizada. Para calcular la vitalidad en cada región del ojo se utiliza el método de *Hamming Distance*¹, que plantea que si dos listas de píxeles son comparadas, dicha distancia es aquel número de píxeles que no tengan el mismo valor.

2.2.2. Método basado en el análisis del enfoque variable

Del autor Sooyeon Kim[19] que se incluye también en el estudio de las texturas de la imagen, descrito en la sección 1.2.3 y que permitirá incluir un mecanismo para detectar la presencia frente a la cámara de imágenes en tres dimensiones.

Procedimiento

En el caso específico de aquellos rostros reales, las regiones enfocadas son claras y otras son borrosas debido a la información de profundidad existente. En cambio en las imágenes tomadas desde una copia impresa de un rostro genuino hay poca diferencia entre la enfocada y la desenfocada. Este método presenta una limitación en cuanto al **Grado de Enfoque** (DoF) que determina el rango para centrarse en variaciones de píxeles de las imágenes secuencialmente tomadas. El DoF es el rango entre los objetos cercanos y los más lejanos para un enfoque determinado. Para aumentar el rendimiento en el proceso de detección de vida, los autores lo han desarrollado centrándose en aquellos valores donde el DoF es menor. En este método se utiliza la **Suma Modificada de Laplace**(SML) para medir el valor del enfoque, ver Figura 2.3. El valor SML representa los grados de enfoque en las imágenes, esos valores se representan como filtro diferencial de una transformación de segundo orden.

En el primer paso es necesario tomar dos fotos secuenciales de la persona que será identificada, es necesario centrarse en la nariz y los oídos como se observa en la Figura 2.4, porque la nariz está lo suficientemente cerca de la lente de la cámara con relación a la distancia de los oídos con respecto a la

¹<http://xlinux.nist.gov/dads/HTML/HammingDistance.html>

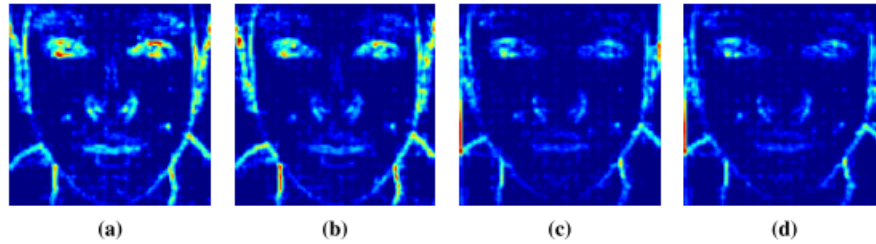


Figura 2.3: SML sobre las imágenes de un rostro vivo en (a) la nariz y (b) las orejas, y SML de imágenes de un rostro no vivo en (c) la nariz y en (d) las orejas.

propia lente. La distancia entre estos elementos del rostro es suficiente para expresar un efecto de tres dimensiones con suficiente orden como para juzgar el grado de enfoque y calcular el valor de SML de ambas imágenes. Posteriormente se calcula la diferencia de SML (DoS) entre ambas imágenes.

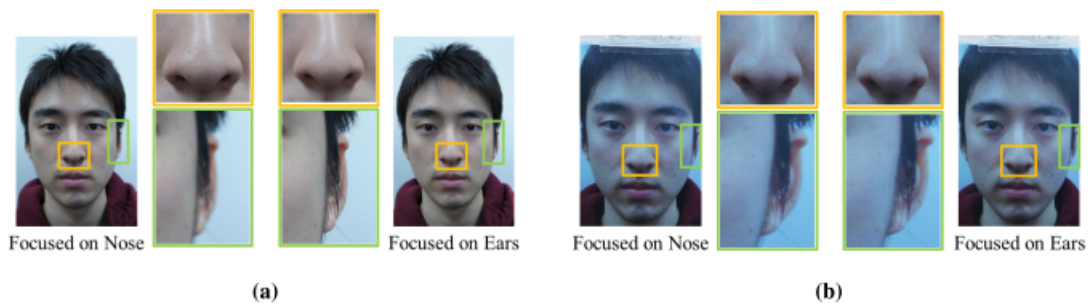


Figura 2.4: Selección de regiones en (a) un rostro real y en (b) una foto.

Características e implementación

Asumiendo que ya se han extraído las regiones a evaluar desde imágenes diferentes, en este caso la región de la nariz y la región de la oreja, se inicia el proceso para calcular los grados de enfoques y luego la diferencia SML_d entre ambas regiones, de la nariz (SML_n) y de la oreja (SML_e).

- Teniendo en cuenta que para esta investigación se utilizaron imágenes tomadas con una webcam común donde el valor del grado de enfoque es reducido, los autores sugieren realizar en este caso un enfoque manual de cada región.

- * El enfoque de forma manual se realiza manipulando la imagen como un vector de dos dimensiones multiplicado por otro vector de dimensión impar, en este caso se utilizó un vector de 3×3 llamado vector de convolución.
 - * El filtro aplicado examina, sucesivamente, cada píxel de la imagen. Para cada uno de ellos, que se llamará "píxel inicial", se multiplica el valor de este píxel y los valores de los 8 circundantes por el valor correspondiente del kernel. Entonces se añade el resultado, y el píxel inicial se regula en este valor resultante final.
 - * En el procedimiento se eliminan o atenúan los componentes de bajas frecuencias en la imagen, resultando en el realce de los bordes y otros detalles marcados o finos de forma que permita obtener un mayor refuerzo de los contornos de los objetos (líneas de cambios bruscos de color o intensidad), lo que proporciona a la imagen filtrada un aspecto de mejora de enfoque.
- Se calcula el grado de enfoque (DoF) en la región de la nariz y en la región de la oreja utilizando la *Suma Laplaciana Modificada*. Este método se basa en el análisis de la vecindad de cada píxel en dependencia del desplazamiento seleccionado en las coordenadas x y y .

Suma de Laplace

$$\boxed{\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}} \quad (2.1)$$

S.K. Nayar en [45] destaca que la segunda derivada de Laplace con respecto a x y con respecto a y pueden tener signos opuestos y provocar que se cancelen una con la otra. El autor propone una modificación utilizando el valor absoluto en cada caso.

Suma de Laplace Modificada

$$\boxed{\nabla_M^2 I = \left| \frac{\partial^2 I}{\partial x^2} \right| + \left| \frac{\partial^2 I}{\partial y^2} \right|} \quad (2.2)$$

La aproximación discreta de Laplace usualmente utiliza un operador de 3×3 , con el objetivo de ajustarse a posibles variaciones de dimensión en la textura de los elementos el autor propone

calcular las segundas derivadas incluyendo una variable de espaciado (*step*) entre los píxeles utilizados para calcular las segundas derivadas.

$$\boxed{\nabla_{ML}^2 I(x, y) = |2I(x, y) - I(x - step, y) - I(x + step, y)| + |2I(x, y) - I(x, y - step) - I(x, y + step)|} \quad (2.3)$$

Finalmente, el grado de enfoque se calcula como la suma de los valores de $\nabla_{ML}^2 I$, en la pequeña vecindad de (x, y) , que sea mayor que el valor de un umbral T .

$$\boxed{F(x, y) = \sum_{i=x-N}^{i=x+N} \sum_{j=y-N}^{j=y+N} \nabla_{ML}^2 I(i, j), \quad \text{para } \nabla_{ML}^2 I(i, j) \geq T} \quad (2.4)$$

- Teniendo los valores de (SML_n) y (SML_e) se determina la diferencia de enfoque.

$$\boxed{SML_d = SML_n - SML_e} \quad (2.5)$$

Para determinar si existe algún nivel de vitalidad se compara entonces la diferencia con un umbral previamente definido.

2.3. Modelo de Dominio

En el modelo de dominio se describen los conceptos fundamentales que son manejados por el componente, lo que permite comprender con mayor facilidad su funcionamiento haciendo referencia a aquellos elementos más significativos dentro de la solución, ver Figura 2.5.

2.3.1. Glosario de conceptos del modelo de dominio

A continuación se definen de forma abreviada los conceptos enunciados en el modelo de dominio:

- **Imagen de Rostro:** Contiene toda la información referente a una imagen facial.
- **Proveedor de Características:** Se encarga de extraer la información de cada rostro y transformarlos de forma tal que puedan ser utilizados por los demás elementos del componente.

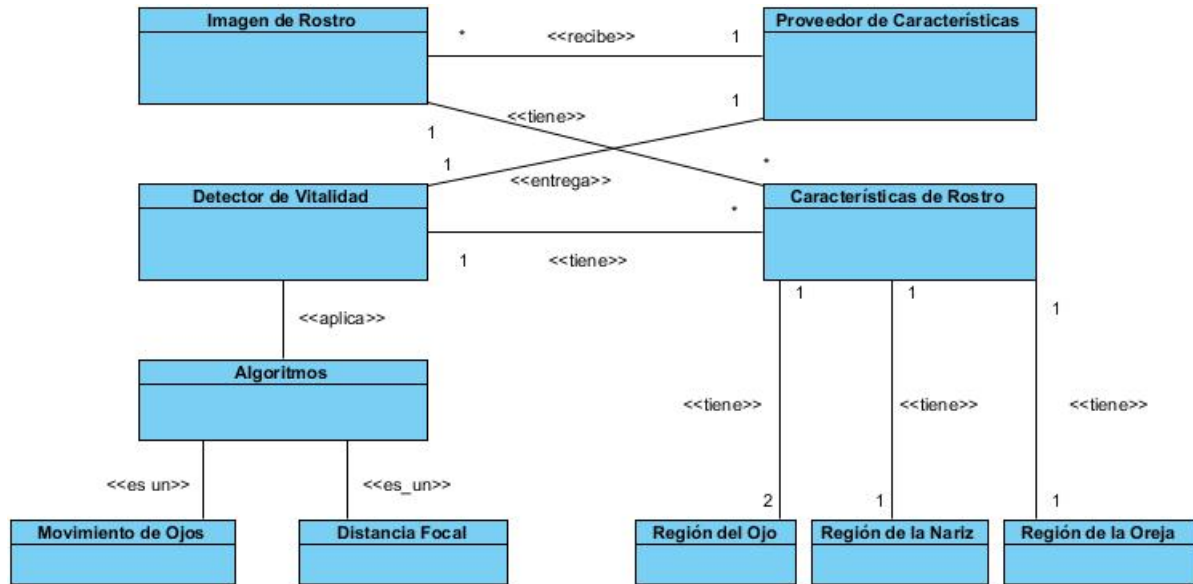


Figura 2.5: Modelo de dominio.

- **Detector de Vitalidad:** Es quien dispone de los procedimientos necesarios para efectuar el análisis sobre el flujo de imágenes, aplica los métodos disponibles y determina el grado de vitalidad según la estrategia definida.
- **Características de Rostro:** Es un concepto generalizador que agrupa todos aquellos elementos identificativos disponibles en la imagen facial.
- **Algoritmo para Movimiento de los Ojos:** Contiene las funciones relacionadas con la detección de vitalidad utilizando como métrica el nivel de movimiento detectado en la región de los ojos.
- **Algoritmo para Distancia Focal:** Contiene las funciones relacionadas con la detección de vitalidad utilizando como métrica la variación en los niveles de enfoque para dos regiones diferentes en un mismo rostro en imágenes consecutivas.
- **Región del Ojo:** Dispone de la información relativa a la región que contiene un ojo específico perteneciente a una imagen facial.
- **Región de la Nariz:** Dispone de la información relativa a la región que contiene la nariz perte-

neciente a una imagen facial.

- **Región de la Oreja:** Dispone de la información relativa a la región que contiene la oreja izquierda perteneciente a una imagen facial vista desde el frente.

2.4. Principales funcionalidades

Los requisitos funcionales de un software son aquellas características que representan lo que el sistema debe hacer. En lo adelante se enumeran aquellos requisitos funcionales definidos para el componente a desarrollar, posteriormente, en la sección 2.6 (Historias de usuario) serán descritos con mayor rigor.

1. Detectar la región del rostro a partir de una imagen.
2. Extraer la región de los ojos a partir de una imagen de rostro.
3. Extraer la región de la nariz a partir de una imagen de rostro.
4. Extraer la región de la oreja a partir de una imagen de rostro.
5. Realizar el proceso de binarizado sobre una imagen.
6. Aumentar el grado de enfoque a una imagen.
7. Detectar el grado de movimiento en la región de los ojos.

- Para satisfacer este requisito es necesario cumplir una serie de condiciones.

Detección: Es necesario detectar la existencia de ojos en cada uno de los rostros extraídos desde la secuencia de imágenes.

Extracción: Teniendo la certeza de que existen regiones de ojo válidas, entonces se extraen dichas regiones y se clasifican según la posición del ojo.

Binarización: Cada región ocular se somete a un procesos de binarización, donde cada píxel es evaluado según su intensidad, asignándole valor blanco o negro según corresponda.

Nivel de movimiento: Teniendo todas aquellas regiones que representan el mismo ojo pero de diferentes imágenes consecutivas, se calcula el grado de variación existente entre ellas. Para calcular la diferencia entre dos imágenes se utiliza el criterio conocido como la *Distancia de Hamming*, el que define realizar una comparación entre las dos imágenes

previamente binarizadas, efectuando píxel a píxel una comparación del tipo XOR, de esta forma se contabilizan aquellos resultados únicamente positivos, resultando así la cantidad de elementos diferentes entre ambas imágenes. Realizando la comparación de cada región de ojo específico con las demás que también pertenecen al mismo ojo dentro de la secuencia, se determina un valor numérico que representa el grado vitalidad de la región analizada.

8. Detectar la variación del grado de enfoque existente en imágenes consecutivas como señal de profundidad.

- Para llevar a cabo este requisito es necesario cumplir con las siguientes actividades.

Detección: Utilizando dos imágenes consecutivas del mismo flujo de video, se extrae la región de la nariz de la primera imagen y utilizando la segunda se extrae la región que enmarca la oreja.

Enfoque manual: A cada región se le aplica un *filtro de alta frecuencia*² para provocar un realce del nivel de enfoque existente en cada área.

Grado de enfoque: Se aplica la Suma Modificada de Laplace a cada una de las imágenes obtenidas anteriormente, de esta forma se determina el grado de enfoque en cada región. El grado de enfoque que se utilizará como resultado será la diferencia entre ambos grados de enfoque, este último valor será tomado como indicador de vitalidad.

2.5. Requisitos no funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Seguidamente se especifican los requisitos no funcionales del sistema a desarrollar.

1. Software:

- Framework de desarrollo: .NET Framework 4.0

²Los filtros de altas frecuencias son los que resaltan o solamente dejan pasar los cambios bruscos de intensidad, mientras que eliminan o minimizan los cambios suaves o degradados. En general, los cambios bruscos de intensidad se producen en los bordes de los objetos. Por esto, los filtros de altas frecuencias son adecuados para resaltar o para individualizar estos bordes. Son conocidos como Sharpening Spatial Filters [46].

- El sistema será usado bajo los sistemas operativos Windows.

2. Hardware:

- PC Intel Pentium 4 o superior.
- CPU 2.13GHZ o superior.
- 512MB de Memoria RAM o superior.

2.6. Historias de usuario

Como define la metodología de desarrollo de software XP para la etapa de implementación de la propuesta de solución, todos aquellos requisitos funcionales y no funcionales son descritos desde las historias de usuario. Utilizando este artefacto (definido por la metodología) el cliente explica las funcionalidades que desea ver en el producto final. A continuación se declaran las historias de usuario que contienen los algoritmos implicados directamente en la detección de vitalidad. (Ver el Anexo A para consultar el resto de las historias de usuario).

Tabla 2.1: HU Detectar el grado de movimiento en la región de los ojos.

Historia de usuario	
Número: CDHU09	Usuario: Sergio Lecier Soler González
Nombre: Detectar el grado de movimiento en la región de los ojos.	
Prioridad de negocio: Alta	Riesgo de desarrollo: Alto
Puntos estimados: 3	Iteración asignada: 2
Programador responsable: Sergio Lecier Soler González	
Descripción: De una secuencia de imágenes se extraen las regiones que representen el mismo ojo en cada una, luego se realiza un proceso de binarizado a cada región para efectuar una comparación utilizando el criterio de la <i>Distancia de Hamming</i> , esta comparación se realiza comparando cada región con las demás y se obtiene un valor numérico que se interpreta como el grado de vitalidad, lo que representa la cantidad de movimiento detectado.	

Observaciones: Se necesita una secuencia de imágenes de rostro de las que se extraerán las características biométricas que necesita el algoritmo, además necesita un valor numérico que representa el umbral para efectuar el proceso de binarizado en cada región de ojo.
Requisitos funcionales: Software
Asuntos pendientes: Ninguno

Tabla 2.2: HU Detectar el grado de enfoque existente en imágenes consecutivas.

Historia de usuario	
Número: CDHU10	Usuario: Sergio Lecier Soler González
Nombre: Detectar el grado de enfoque existente en imágenes consecutivas.	
Prioridad de negocio: Alta	Riesgo de desarrollo: Alto
Puntos estimados: 3	Iteración asignada: 2
Programador responsable: Sergio Lecier Soler González	
Descripción: De dos imágenes consecutivas se extrae la región de la nariz de la primera y la región que contiene la oreja en la segunda, se realiza un reajuste manual del grado de enfoque y se evalúa el nivel de enfoque resultante en cada caso, posteriormente se calcula la diferencia de enfoque entre ambas regiones y se obtiene un valor numérico. El valor obtenido se compara con un umbral y se interpreta como un comportamiento de volumen debido a la diferencia entre la distancia focal de la nariz y la región de la oreja.	
Observaciones: Se necesita una secuencia de imágenes, como mínimo dos que hayan sido extraídas de un flujo de video, debe ser posible detectar la presencia de un rostro en cada caso para extraer de cada una las características biométricas necesarias para efectuar la evaluación.	
Requisitos funcionales: Software	
Asuntos pendientes: Ninguno	

2.7. Planificación

La planificación en la metodología XP se realiza de acuerdo a la relación que existe entre el cliente y los desarrolladores. Los clientes definen las necesidades propias del negocio como pudieran ser el

alcance, la prioridad con que se implementen las historias de usuario y los períodos entre cada fecha de entrega del producto final. Por otro lado los programadores se enfocan en los requerimientos técnicos que también inciden sobre las necesidades del negocio, por ejemplo, estiman la duración necesaria para implementación de cada historia de usuario y la organización del proceso de desarrollo en general.

2.7.1. Plan de entrega

De acuerdo con la metodología se determina el cronograma en función de la prioridad definida por el cliente con respecto a las historias de usuario. Se han definido 2 iteraciones en las que se codificarán 10 historias de usuario.

Tabla 2.3: Plan de entrega.

Entregable	Iteración	Fin de iteración
Herramientas de procesamiento de imágenes	1	febrero de 2015
Esquema de detección de vitalidad	2	abril de 2015

2.7.2. Plan de iteraciones

Con el objetivo de obtener un prototipo funcional al concluir cada iteración, se ha definido un plan de iteraciones teniendo en cuenta el orden en que serán implementadas las historias de usuario descritas.

Tabla 2.4: Plan de iteraciones.

Iteración	No. HU	Historia de usuario	Semanas estimadas
1	CDHU01	Detectar un rostro a partir de una imagen.	4
	CDHU02	Extraer la región de los ojos.	
	CDHU03	Extraer la región de la nariz.	
	CDHU04	Extraer la región de la oreja.	
	CDHU05	Realizar el proceso de binarizado sobre una imagen.	
	CDHU06	Aumentar el grado de enfoque a una imagen.	

	CDHU07	Calcular la Distancia de Hamming entre dos imágenes binarizadas.	
	CDHU08	Calcular la Suma Modificada de Laplace a partir de una imagen.	
2	CDHU09	Detectar el grado de movimiento en la región de los ojos.	4
	CDHU10	Detectar el grado de enfoque existente en imágenes consecutivas.	

En la primera iteración se implementarán aquellos requisitos relacionados con la detección de características biométricas a partir de una imagen facial, además, serán desarrolladas las herramientas necesarias para el posterior procesamiento de las imágenes. Mientras que la segunda iteración estará orientada al desarrollo de aquellas funcionalidades implicadas en el proceso de detección de vitalidad.

2.8. Diseño

En el diseño se indica como se construirá la propuesta de solución, para ello se definirá una estructura de la forma más sencilla posible, que sea capaz de satisfacer los requisitos funcionales y no funcionales anteriormente descritos. Se identifican además todos aquellos elementos que componen la estructura del software, atributos y funciones.

2.8.1. Tarjetas CRC

El uso de las tarjetas CRC está definido en la metodología de desarrollo seleccionada con el objetivo de mantener el paradigma orientado a objetos. Las siglas CRC son el acrónimo en inglés de *Class*, *Responsibilities* y *Collaborators* que son los componentes que se describen en cada tarjeta. A continuación se definen las tarjetas CRC de las principales clases identificadas, para consultar el resto ver el Anexo B de Tarjetas CRC.

Tabla 2.5: Tarjeta CRC Detector de Vitalidad.

Detector de Vitalidad	
Responsabilidades	Colaboradores

Extraer características biométricas	Proveedor de características
Detectar vitalidad usando volumen	Detector de volumen
Detectar vitalidad usando movimiento	Detector de movimiento
Detectar vitalidad usando método combinado	Detector de volumen, Detector de movimiento

Tabla 2.6: Tarjeta CRC Proveedor de Características.

Proveedor de Características	
Responsabilidades	Colaboradores
Obtener listado de regiones de ojo	Imagen de Rostro, Herramientas de imagen
Obtener listado de regiones de ojo binarizados	Imagen de Rostro, Herramientas de imagen
Extraer región de la nariz	Imagen de Rostro, Verificador de imagen
Extraer región de la oreja	Imagen de Rostro, Verificador de imagen

2.8.2. Diagrama de clases del diseño

Al concluir la definición de las tarjetas CRC han quedado identificadas las clases del sistema y sus principales funcionalidades, en este momento se está en condiciones de confeccionar el diagrama de clases del diseño que aunque no es un artefacto propio de la metodología XP permitirá representar gráficamente las clases (atributos y métodos), las relaciones entre clases (dependencias y asociaciones) y la navegabilidad.

En el diagrama de clases del diseño, ver Figura 2.6, se puede apreciar la estructura del componente así como la relación que existe entre sus funcionalidades internas.

Funcionalidades expuestas a los sistemas externos

El componente en su funcionamiento es totalmente independiente y se propone una arquitectura capaz de incorporar otros mecanismos de detección de vitalidad sin provocar afectaciones en la estructu-

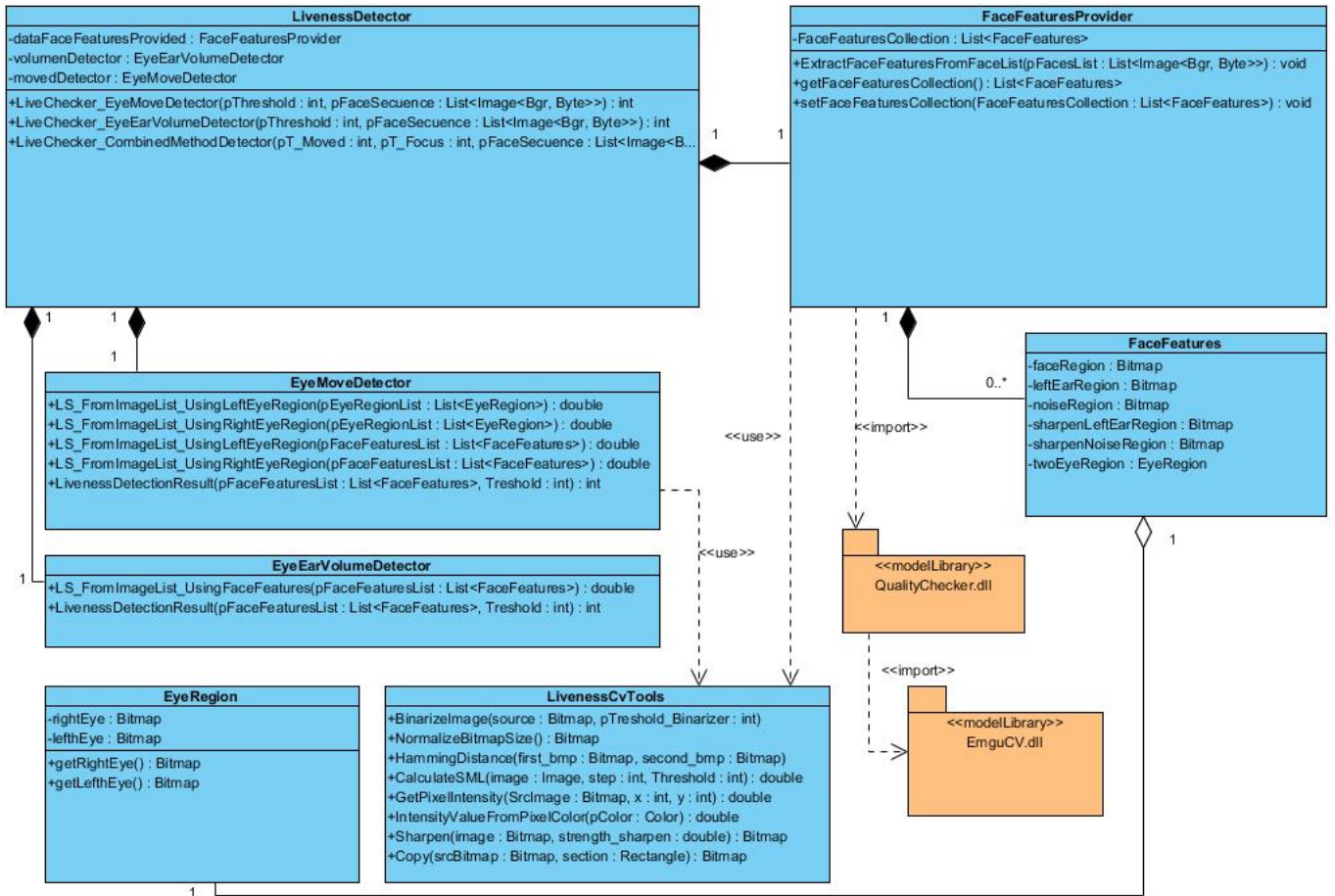


Figura 2.6: Diagrama de clases del diseño.

ra principal. Para utilizar el componente propuesto es necesario conocer las funcionalidades que han quedado expuestas a los sistemas que deseen realizar detección de vitalidad en su proceso de captura de imágenes, dichos sistemas deben utilizar la clase *LivenessDetector* que brinda las siguientes funcionalidades:

- **LiveCheckerEyeMoveDetector:** Evalúa la vitalidad utilizando únicamente el análisis de movimiento detectado en la región de los ojos.
- **LiveCheckerEyeEarVolumeDetector:** Evalúa la vitalidad utilizando únicamente el análisis del grado de enfoque detectado en las imágenes.

- **LiveCheckerCombinedMethodsDetector** Realiza una combinación de los dos métodos anteriores.

En los primeros dos casos se recibe una lista de imágenes que contienen el rostro a evaluar y el umbral a utilizar en la comparación y en el último caso además de la secuencia de imágenes también se deben especificar los dos umbrales por separado.

2.8.3. Patrones de diseño

Los patrones de diseño son esquemas que dan solución a problemas similares que comúnmente ocurren durante el diseño de un software.

Los patrones GRASP (General Responsibility Assignment Software Patterns) describen los principios fundamentales de la asignación de responsabilidades a objetos. Los patrones de este tipo usados son:

- **Controlador:** Es un intermediario entre la capa de procesamiento de imágenes y la capa de detección de vitalidad, coordina la actividad entre los objetos. Por ejemplo, en el sistema se hace uso del patrón controlador definiendo la clase *LivenessDetection*, que gestiona el flujo de información entre ambas capas.
- **Experto:** Establece que cada responsabilidad sea asignada a la clase contenedora de la información necesaria para cumplirla. Garantiza el mínimo de relaciones entre las clases, para así obtener un sistema sólido y fácil de mantener; favorece el bajo acoplamiento y la alta cohesión. Por ejemplo, este patrón se pone de manifiesto en la clase *EyeMoveDetector* que contiene de forma exclusiva toda la información para detectar movimiento en la región de los ojos.
- **Creador:** Establece que una instancia de un objeto solo pueda ser creada por la clase que contiene la información necesaria para ello. Su uso permite crear las dependencias mínimas necesarias entre las clases, beneficiando el mantenimiento del sistema y brindando mejores oportunidades de reutilización; favorece el bajo acoplamiento. Por ejemplo, este patrón se pone de manifiesto en la clase *FaceFeaturesProvider* que contiene (en realidad, agrega) muchos objetos *FaceFeatures*, por lo que el patrón mencionado sugiere que la primera sea la encargada de asumir la responsabilidad de crear las instancias de la segunda.

- **Alta cohesión:** Basado en la asignación de responsabilidades teniendo en cuenta que la cohesión permanezca alta. Su utilización facilita la comprensión del diseño e incrementa las capacidades de reutilización.
- **Bajo acoplamiento:** Plantea que debe existir una alta reutilización entre las funcionalidades de las clases con una mínima dependencia, contribuyendo así al mantenimiento de las mismas. El empleo de los patrones Experto y Creador favorecen al bajo acoplamiento entre las clases del sistema.

Los patrones GoF (Gang of Four) describen las formas en las que pueden ser organizados los objetos para trabajar unos con otros, formando estructuras de mayor complejidad. El patrón de este tipo utilizado es:

- **Singleton:** Está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto.
 - La propia clase es responsable de crear la única instancia.
 - Permite el acceso global a dicha instancia mediante un método de clase.
 - Declara el constructor de clase como privado para que no sea instanciable directamente.

Por ejemplo, en la solución este patrón se evidencia específicamente en la definición de la clase *LivenessCvTools* que contiene un grupo de filtros para el tratamiento de imágenes, para su uso no es necesario emitir una nueva instancia de la clase, solo utilizar la única instancia que provee este patrón y acceder a las funcionalidades que brinda.

2.9. Arquitectura

La arquitectura es la representación global de la estructura de un software, en ella se definen los componentes funcionales y la relación existente entre ellos. Describe además la aplicación de un conjunto de patrones y condiciones que supervisan su estructura, se especifica la forma en que se utilizan los estándares, convenciones, reglas y procesos [47].

En el desarrollo del componente se utilizará una arquitectura organizada en capas la cual responde al estilo arquitectónico de llamada y retorno. Este tipo de distribución permite el desarrollo de la

aplicación de forma modular, lo que facilita la corrección de errores, además posibilita brindar un mejor soporte al sistema una vez finalizado su desarrollo. Utilizando este enfoque se obtiene un sistema escalable en gran medida, donde la inclusión futura de otros métodos para la detección de vitalidad no incurra en cambios estructurales entre las funcionalidades ya existentes [48].

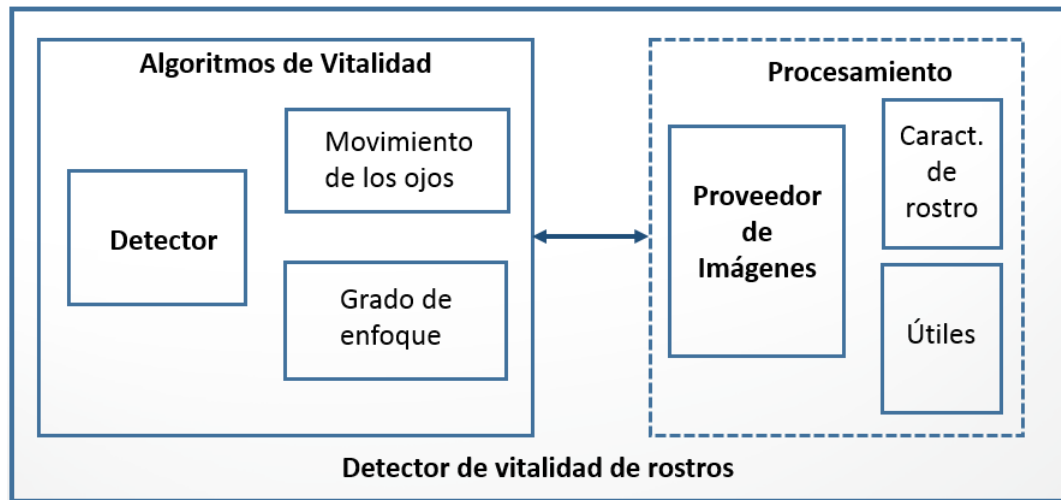


Figura 2.7: Arquitectura del componente.

2.9.1. Patrón arquitectónico

En la solución se propone el uso de la arquitectura en dos capas, en una capa se encuentran todas las herramientas para el procesamiento digital de imágenes y la extracción de aquellas características biométricas necesarias; por su parte en la capa restante estarán agrupadas aquellas funciones asociadas al proceso de detección de vitalidad, dígase el procedimiento general de los algoritmos y sus funciones de evaluación (Ver Figura 2.7).

2.10. Conclusiones parciales

En este capítulo se expusieron las características del sistema después del análisis de los elementos que constituyen el componente para la detección de vitalidad. Se definió el modelo de dominio que ayudó a identificar los requisitos funcionales y no funcionales descritos en las historias de usuario.

La solución propuesta implementa una arquitectura n-capas, específicamente se definen 2 capas: *Algoritmos de Vitalidad* y *Procesamiento de Imágenes*. El diseño está orientado a la extensibilidad pues

permite que el sistema a desarrollar se le puedan incluir en el futuro otros mecanismos para la detección de vitalidad sin que provoque cambios a la estructura principal, además, se describe la forma en que la propuesta de solución cumple con los patrones de diseño GRASP y GoF, estos últimos son responsables de garantizar la organización de las clases y la estructura de trabajo.

Capítulo 3: Implementación y pruebas.

3.1. Introducción

En el presente capítulo se describe la implementación del software, fase donde se materializa el producto final y se cumple con los requisitos obtenidos al inicio de la investigación. Para lograr esto se generan los diagramas de componentes y de despliegue, donde se observan las dependencias lógicas entre los elementos de software. Además se describe el funcionamiento de los métodos de detección utilizados y la forma en que se deben definir los umbrales para el criterio de decisión con respecto al grado de vitalidad detectado en cada caso. Finalmente se muestran las pruebas realizadas para validar el funcionamiento correcto del componente.

3.2. Tareas de ingeniería

Utilizando las historias de usuario se define el trabajo de cada iteración y se expresa mediante las tareas de ingeniería necesarias para dar cumplimiento a los objetivos en cada momento. Cada tarea es asignada a un programador directamente, pero puede ser realizada por una pareja de programadores según la metodología de desarrollo seleccionada. En las historias de usuario no aparece toda la descripción necesaria para la implementación, por lo que las tareas de ingeniería las complementan e indican a los programadores las acciones a realizar en cada una de ellas.

A continuación se muestran las tareas de ingeniería implicadas en la detección de vitalidad, ver el Anexo C para consultar el resto de las tareas de ingeniería.

Tabla 3.1: Tareas de ingeniería de la segunda iteración.

Iteración 2	
Historia de usuario	Tareas

<p>Detectar el grado de movimiento en la región de los ojos</p>	<ol style="list-style-type: none">1. Obtener una secuencia de imágenes.2. Detectar el rostro en cada imagen.3. Extraer la región que representa el mismo ojo de cada rostro detectado.4. Normalizar las dimensiones y binarizar cada región.5. Calcular el grado de movimiento utilizando la <i>Distancia de Hamming</i>.
-----------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p>Detectar el grado de enfoque existente en imágenes consecutivas.</p>	<ol style="list-style-type: none"> 1. Obtener una secuencia de imágenes. 2. Utilizando dos imágenes consecutivas a partir de la secuencia, es necesario detectar el rostro en cada una. 3. Utilizando la primera imagen, extraer la región que contiene la nariz. 4. Utilizando la segunda imagen, extraer la región que contiene oreja. 5. Realizar un reajuste del grado de enfoque en las dos regiones anteriores. 6. Calcular el grado de enfoque en cada región, se utilizará la Suma Modificada de Laplace para efectuar la operación sobre la imagen en cada caso. 7. Calcular la diferencia de enfoque entre ambas regiones.
-------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3.3. Estándares de codificación

La metodología de desarrollo de software XP promueve la propiedad colectiva del código, por lo que establecer un estándar para la codificación que sea de conocimiento general es de vital importancia, además facilita la constante y necesaria refactorización del propio código. Esta práctica también favorece el mantenimiento de la aplicación en caso de ser necesario añadir funcionalidades nuevas, modificar o mejorar el rendimiento del sistema en general.

3.3.1. Estilos para la capitalización de los identificadores

Para la capitalización de los identificadores se utilizaron los convenios:

- **Pascal:** La primera letra en el identificador y la primera letra de cada subsiguiente palabra concatenada se capitalizan. Por ejemplo: `ExtractLeftEye`.
- **Camel:** La primera letra en el identificador se pone en minúscula y la primera letra de cada subsiguiente palabra concatenada en mayúscula. Por ejemplo: `eyeRegionList`.

La convención Pascal se empleó en los identificadores de las clases, métodos y nombres de ficheros. Mientras que la convención Camel es el estilo de los identificadores de las variables, atributos y parámetros.

En los identificadores de las variables se tuvo en cuenta emplear su significado para nombrarlos y no utilizar las abreviaturas. Los nombres de los métodos están en correspondencia con la funcionalidad que realizan, los ficheros incluidos en la solución son identificados por el nombre de la única clase que contienen.

3.3.2. Comprensión y legibilidad del código

Con el objetivo de hacer más legible el código se insertaron comentarios en todas las declaraciones de las clases y en aquellas funciones de mayor complejidad. Los atributos, propiedades y constructores aparecen en la parte superior de las clases, mientras que los métodos privados y públicos aparecen a continuación.

3.4. Diagrama de componentes

El diagrama de componentes, como indica su nombre, representa la división del software en partes atómicas denominadas componentes. Este modelo muestra la organización y las dependencias entre los distintos elementos que integran el software. Las dependencias señaladas en este diagrama indican que un componente utiliza las funcionalidades ofrecidas por otro componente adyacente (Ver Figura 3.1).

3.4.1. Descripción del diagrama de componentes

En el diagrama se muestran las partes que conforman la solución según las responsabilidades dentro del componente.

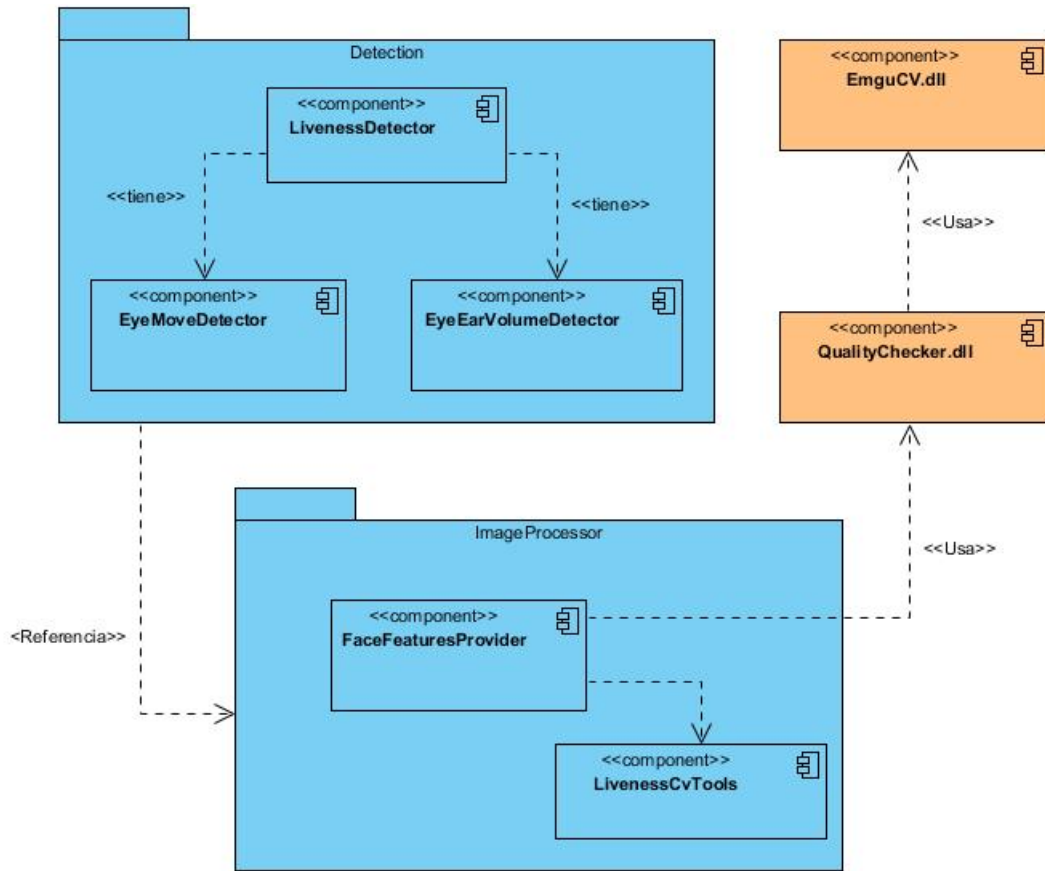


Figura 3.1: Diagrama de componentes.

En el paquete **Detection** se encuentra el componente **LivenessDetector** que es el encargado de realizar las operaciones de detección de vitalidad haciendo uso de los algoritmos disponibles desde los componentes **EyeMoveDetector** y **EyeEarVolumeDetector**.

Desde el paquete **Detection** se utilizan otras funcionalidades que están disponibles tan solo en el paquete **ImageProcessor**, este último es el encargado de extraer las características biométricas y realizar el procesamiento previo de las imágenes. Utiliza el componente **LivenessCvTools** para realizar operaciones sobre las imágenes y en el proceso de extracción de características utiliza los componentes externos **QualityChecker** y **EmguCV** respectivamente.

3.5. Método para la detección de movimiento en los ojos

Este enfoque fue analizado en la Sección 1.2.2 y luego descrito su funcionamiento en la Sección 2.2.1, está basado en el análisis de los movimientos involuntarios que ocurren en la región ocular de cada persona, dígame pestañeo, cambios en la dirección de la vista, etc. Los autores que proponen este enfoque afirman que los movimientos ocurren con una frecuencia que permite detectar pequeños cambios en la región que ocupa cada ojo y zonas cercanas a partir de una imagen facial obtenida desde una secuencia de video [18].

3.5.1. Definición del umbral de decisión

La detección de vitalidad en este método está basado en el análisis del grado de movimiento detectado, que se calcula evaluando la variación existente entre las regiones que representan el mismo ojo de cada rostro. Para tomar una decisión en cuanto a la vitalidad es necesario comparar el valor numérico que representa la cantidad de cambios detectados con un umbral lo suficientemente cercano al punto donde la toma de decisiones genere la relación deseada entre los indicadores de *FRR* y *FAR*.

El valor asignado al umbral estará estrechamente relacionado con la naturaleza y calidad de las imágenes utilizadas, esto se define según la resolución de cada imagen en consecuencia con las propiedades del hardware empleado para la captura de las fotos, estos elementos a tener en cuenta permitirán extraer en mayor o menor medida las características de cada imagen. Tomando como referencia esta última afirmación se debe definir un umbral que se corresponda con las características de cada entorno donde se implemente este mecanismo de detección de vitalidad.

3.5.2. Calcular el umbral de comparación

A continuación se define el procedimiento para obtener los valores de falso rechazo, para determinar este indicador las fotos utilizadas fueron tomadas a individuos reales.

- El procedimiento se inicia seleccionando N conjuntos de igual dimensión que contengan fotos secuenciales de individuos diferentes, pueden ser tomadas utilizando un flujo de video.
- Se calcula para cada conjunto X_i el grado de vitalidad (LS). Se utiliza el criterio de la media de la suma de todas las distancias obtenidas haciendo una comparación de todos con todos, en esta investigación se utilizaron 10 imágenes reales de cada individuo, lo que indica tener que realizar 100 comparaciones para obtener cada $LS(X_i)$.

- En este momento se tiene el valor de LS obtenido para cada conjunto, se requiere hacer el cálculo de la media entre todos los indicadores de vitalidad $LS(X_i)$.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^N LS(X_i) \quad (3.1)$$

- Calcular posteriormente la varianza σ^2 respecto a la media \bar{x} de los valores obtenidos para cada secuencia de rostros.

$$\sigma^2 = \frac{\sum_{i=1}^N X_i^2}{N} - \bar{x}^2 \quad (3.2)$$

- Se define el intervalo utilizando la desviación típica según los valores obtenidos.

$$[(\bar{x} - \sigma), (\bar{x} + \sigma)] \quad (3.3)$$

- Utilizando el valor de LS para cada secuencia X_i y estableciendo un valor α como paso, se verifica cuántos elementos de la muestra de tamaño N permanecen dentro del intervalo partiendo desde el valor $(\bar{x} - \sigma)$ con paso α hasta $(\bar{x} + \sigma)$.

* Para cada iteración se relaciona el valor del umbral establecido temporalmente según el paso α con la cantidad de $LS(X_i)$ que permanecen dentro del intervalo, de esta forma se puede describir el comportamiento del método propuesto en relación a la tasa de falso rechazo teniendo en cuenta la calidad de las imágenes, la respuesta obtenida utilizando imágenes genuinas y los posibles valores para el umbral.

En el procedimiento descrito anteriormente se hace referencia a imágenes reales por lo que el valor obtenido se refiere al criterio de falso rechazo (FRR), este procedimiento también debe realizarse con imágenes falsas hechas a partir de fotos de individuos reales para obtener las tasas de falsa aceptación (FAR), el punto donde las gráficas se corten resulta el valor mínimo para ambos indicadores. La selección del umbral debe estar en correspondencia con el nivel de tolerancia con respecto a cada indicador. En la Figura 3.2 se muestra la gráfica resultante.

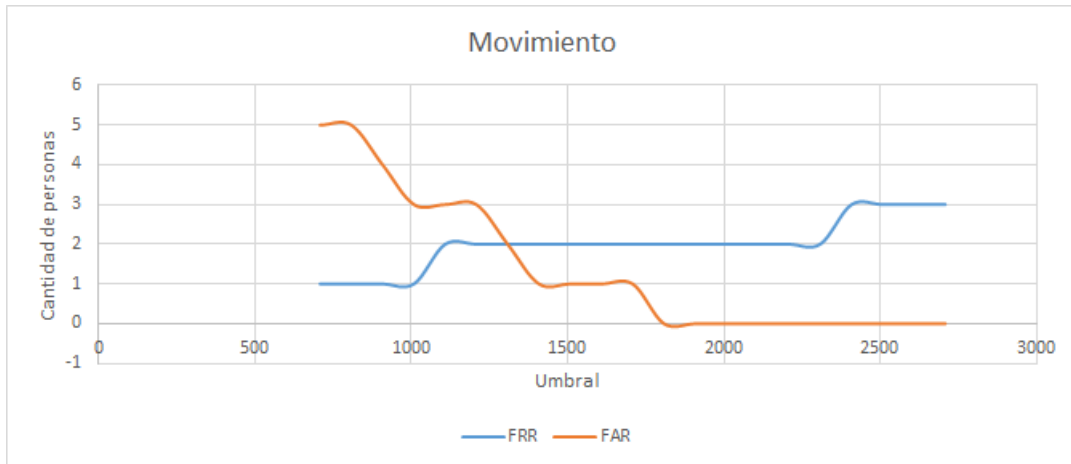


Figura 3.2: Resultados de la definición del umbral de movimiento.

3.6. Método para la detección de volumen

Este método fue analizado en la Sección 1.2.3 y su funcionamiento descrito en la Sección 2.2.2, se basa en la relación que existe al enfocar una cara real y otra que no lo es, el procedimiento se realiza evaluando dos imágenes consecutivas extraídas de un mismo flujo de video y evaluando los niveles de enfoque de cada región analizada, los autores afirman que si se trata de una falsificación la diferencia de enfoque entre ambas regiones debe ser mínima [19].

3.6.1. Calcular el umbral de comparación

El umbral de decisión se determina aplicando un procedimiento similar al descrito en la Sección 3.5.2, en este caso se sustituye el valor de la diferencia del grado de enfoque ($SML_d(X_i)$) por el criterio de $LS(X_i)$ utilizado en el método anterior.

De igual forma se define una cantidad N de muestras verdaderas de rostros y se inicia la operación para determinar el intervalo y luego las tasas de FAR y FRR, lo que permitirá establecer un criterio numérico como umbral de comparación. En la Figura 3.3 se muestra la gráfica resultante.

3.7. Pruebas

Cuando se va a medir la calidad de un producto es necesario tener en cuenta las pruebas realizadas en función del grado de satisfacción de los requisitos definidos. El principal objetivo es detectar los

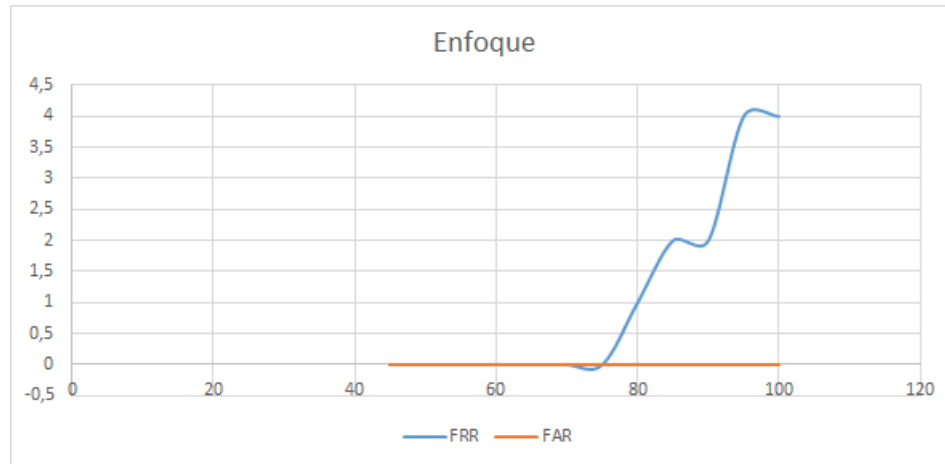


Figura 3.3: Resultados de la definición del umbral de enfoque.

errores y la metodología XP define dos tipos de pruebas: *pruebas unitarias* y *pruebas de aceptación o funcionales*. Además con el objetivo de comprobar la eficiencia en el comportamiento del componente durante el procesamiento y obtener aquellos indicadores que permitieran establecer una métrica para evaluar el mismo, se le realizaron *pruebas de rendimiento*.

3.7.1. Pruebas unitarias

Las pruebas unitarias son escritas por los programadores antes de comenzar la implementación. El objetivo de estas pruebas es aislar pequeñas e individuales porciones del código para verificar que no tengan errores. Para la realización de estas pruebas se utilizó un Add-in del entorno de desarrollo utilizado llamado ReSharper(R#)¹ que permite detectar errores y brinda soluciones inmediatas para corregirlos. Las Figuras (3.4) y (3.5) muestran el resultado de dos pruebas, ver Anexo D para consultar el resto de las pruebas realizadas.

3.7.2. Pruebas de aceptación

Las pruebas de aceptación o pruebas funcionales son definidas por el cliente y son esenciales para su satisfacción con el producto desarrollado. Constituyen el fin de una iteración y el comienzo de la siguiente que asumirá la corrección de los errores encontrados.

Al sistema se le realizaron dos tipos de pruebas de aceptación: las pruebas alfa y las pruebas beta.

¹<https://www.jetbrains.com/resharper/>

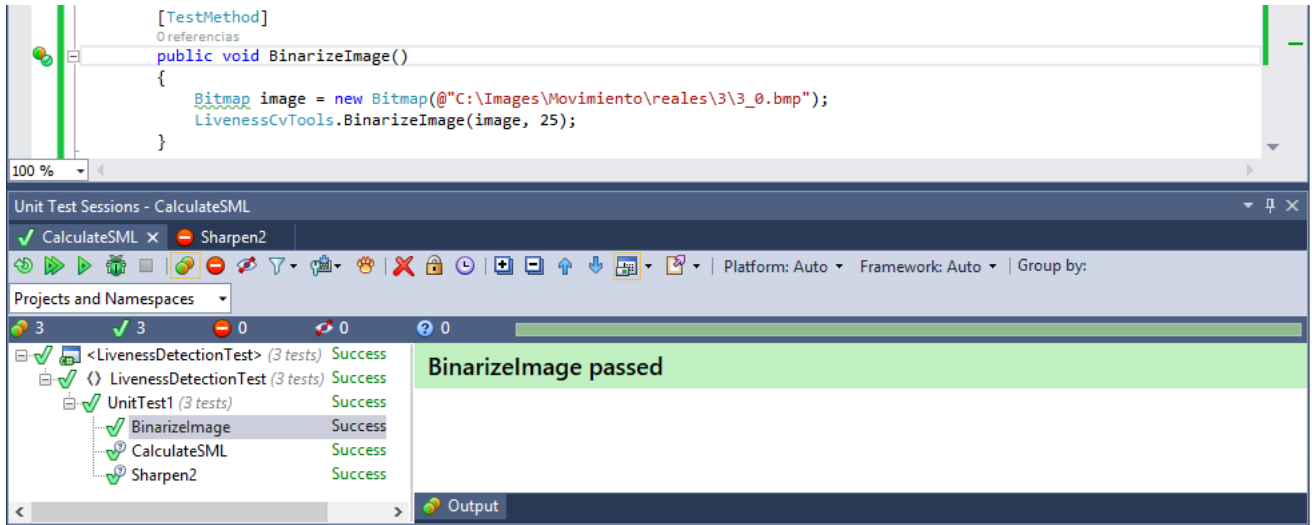


Figura 3.4: Función para binarizar una imagen.

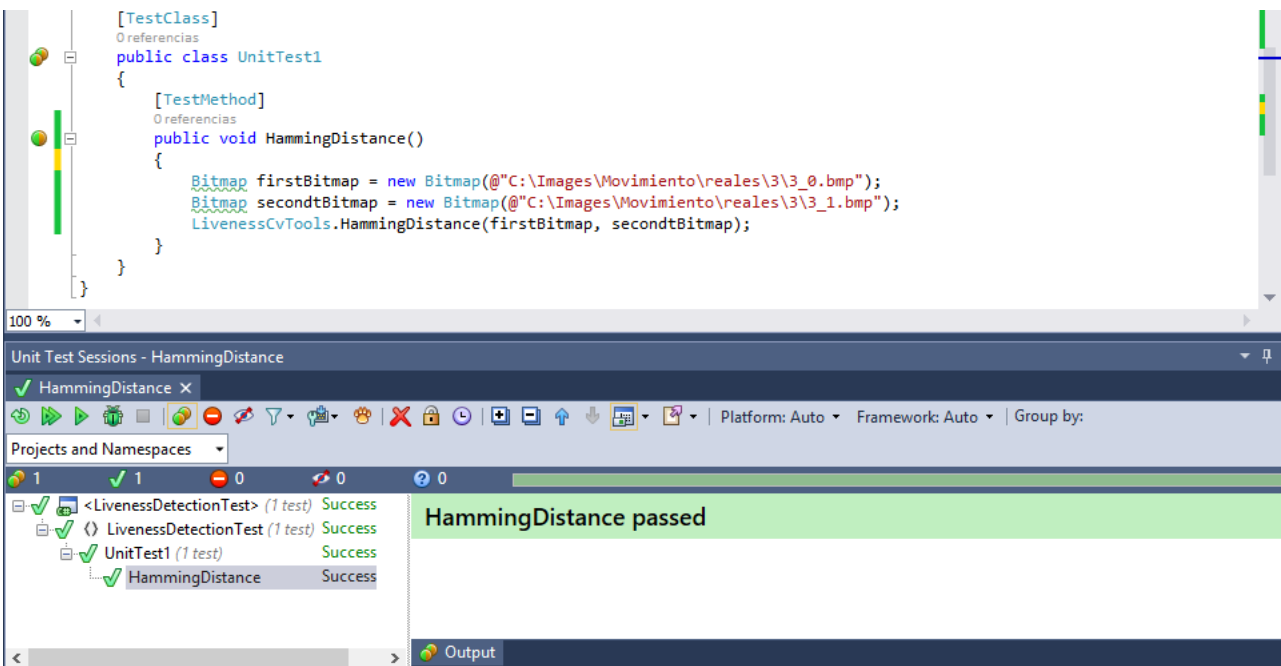


Figura 3.5: Función para calcular la distancia de hamming.

Las pruebas alfa fueron realizadas a las historias de usuario implementadas en la primera iteración, en esta etapa el sistema solo disponía de las funcionalidades necesarias para el tratamiento de imágenes.

Cuando se completó la segunda iteración y el sistema disponía de las funcionalidades de verificación de vitalidad se realizaron las pruebas betas para comprobar su comportamiento.

A continuación se describen los casos de pruebas relacionados con la detección de vitalidad de la segunda iteración, se recomienda consultar el Anexo E con las pruebas alfa.

Tabla 3.2: CP Detectar movimiento en la región del ojo.

Caso de prueba de aceptación	
Código: CDHU09CP9	HU: Detectar el grado de movimiento en la región de los ojos
Responsable de la prueba: Sergio Lecier Soler González.	
Descripción: Prueba para verificar que dado una secuencia de imágenes faciales se pueda detectar el grado de vitalidad.	
Condiciones de ejecución: El componente debe recibir una secuencia de imágenes de rostros válidos, de donde se puedan extraer las características biométricas necesarias para la ejecución.	
Entrada/Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Ejecutar la función pasándole la secuencia de imágenes y el umbral como parámetros. 2. Obtener la respuesta. 	
Resultado esperado: Utilizando una secuencia de imágenes tomadas utilizando rostros genuinos, el componente debe aceptar el rostro como válido.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 3.3: CP Detectar el grado de enfoque existente en imágenes consecutivas.

Caso de prueba de aceptación	
Código: CDHU10CP10	HU: Detectar el grado de enfoque existente en imágenes consecutivas
Responsable de la prueba: Sergio Lecier Soler González.	
Descripción: Prueba para verificar que dado dos imágenes faciales consecutivas se pueda determinar la diferencia del grado de enfoque entre ellas e interpretar el resultado como la existencia de volumen en la imagen.	

<p>Condiciones de ejecución: El componente debe recibir una secuencia de imágenes de rostros válidos para seleccionar dos de ellas y que se puedan extraer las características biométricas necesarias para la ejecución.</p>
<p>Entrada/Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. Ejecutar la función pasándole la secuencia de imágenes y el umbral como parámetros. 2. Obtener la respuesta.
<p>Resultado esperado: Utilizando una secuencia de imágenes tomadas utilizando rostros genuinos, el componente debe aceptar el rostro como válido.</p>
<p>Evaluación de la prueba: Prueba satisfactoria.</p>

Tabla 3.4: CP Verificar la integración del componente con el Sistema de Reconocimiento Facial (SRF).

Caso de prueba de integración
<p>Componentes que se integran: El SRF y el Componente para la Detección de Vitalidad.</p>
<p>Descripción: Esta prueba está orientada a verificar el correcto funcionamiento luego de efectuar la inclusión de la propuesta de solución en un SRF.</p>
<p>Condiciones de ejecución: El SRF debe incluir en su negocio la detección de vitalidad.</p>
<p>Entrada/Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. El SRF debe efectuar la captura de imágenes secuenciales de un rostro con vitalidad. 2. Transferir la secuencia de imágenes al componente de detección en el formato definido utilizando la función expuesta para la detección. 3. El SRF debe esperar hasta obtener la respuesta relacionada con la detección de vitalidad.
<p>Resultado esperado: El componente debe retornar el valor de estado (1) que se interpreta como respuesta de rostro con vida.</p>
<p>Evaluación de la prueba: Prueba satisfactoria.</p>

3.7.3. Pruebas de rendimiento

Las pruebas de rendimiento son en general aquellas realizadas para determinar cómo un sistema se desempeña en términos de capacidad de respuesta y estabilidad del funcionamiento bajo una carga de trabajo particular. También pueden servir para investigar, medir, validar o verificar otros atributos de calidad del sistema, tales como la escalabilidad, la fiabilidad y el uso de recursos [49].

En esta investigación fueron incluidas con el objetivo de evaluar la fiabilidad existente en las respuestas emitidas por el componente al concluir el proceso de detección de vitalidad. Teniendo en cuenta que en el componente desarrollado se utilizaron dos métodos, en consecuencia se diseñaron las pruebas por separado, *detección de movimiento* y *evaluación del grado de enfoque*.

Pruebas a la detección de movimiento

Para comprobar el margen de desempeño de este método, se construyó una base de datos local con imágenes capturadas utilizando una webcam de uso común. En el proceso de captura participaron diez individuos y de cada uno se tomaron 10 imágenes consecutivas, posteriormente se imprimieron las fotos y se repitió la captura de imágenes utilizando entonces las fotos de cada individuo. En total se obtuvieron 100 imágenes reales y 100 imágenes falsas.

La implementación propuesta arrojó resultados alentadores con un índice del 80 % de aceptación, 10 % de FRR y 10 % de FAR. El umbral utilizado se obtuvo realizando el procedimiento con las secuencias de imágenes reales y falsas de los 10 individuos como se describió en la sección 3.5.2. Ver la Figura 3.6 para más detalles de la prueba.

Pruebas a la evaluación del grado de enfoque

Para las pruebas realizadas a este método se utilizaron imágenes de la base de datos *NUAA Imposter Database*, se seleccionaron 10 individuos y luego se escogieron dos imágenes válidas y dos falsas de cada uno, para un total de 40 imágenes.

Los resultados obtenidos indican un 90 % de aceptación para un 10 % de FRR, en este caso también se definió el umbral según la metodología descrita en la sección 3.5.2. En la Figura 3.6 se describen los resultados obtenidos.

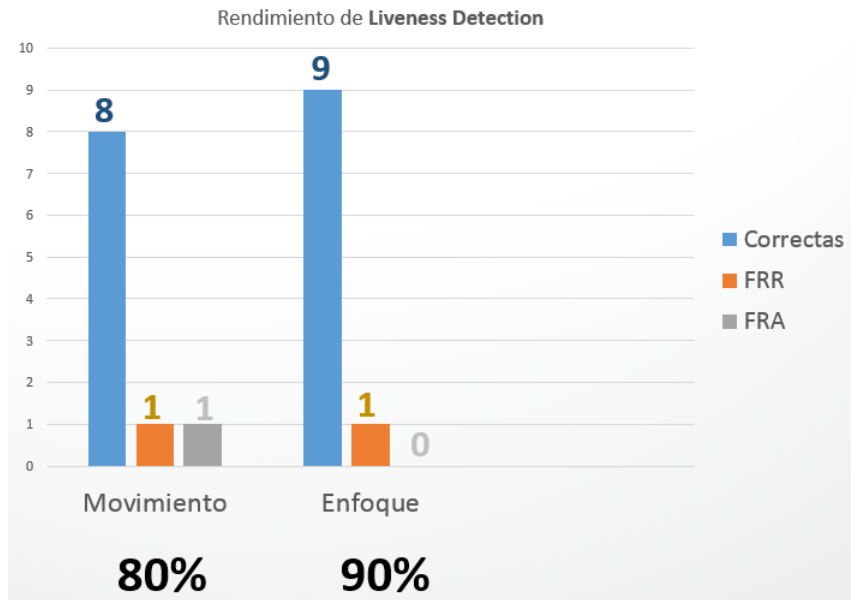


Figura 3.6: Resultados de las pruebas de rendimiento.

3.8. Conclusiones parciales

En este capítulo se describió el tipo de codificación y el tipo de pruebas a las que fue sometido el sistema. Las labores de ingeniería ayudaron en la codificación del componente, facilitando la comprensión de las funcionalidades a desarrollar. El diagrama de componentes describió como interactúan los paquetes de clases y las diferentes librerías utilizadas. Se constató que el desarrollo guiado por pruebas asegura la ejecución correcta de la solución en todo el período de implementación, disminuyendo el tiempo invertido en el ciclo compilación-ejecución.

Las pruebas de aceptación realizadas demostraron la satisfacción del cliente en cuanto al funcionamiento del proceso de detección de vitalidad. Analizando los resultados obtenidos de las pruebas de rendimiento, se comprobó que la solución propuesta reportó resultados alentadores, satisfaciendo el objetivo general de la investigación, aunque se detectó que los resultados del análisis de las texturas pudiera estar condicionado por la información de ruido existente en las imágenes utilizadas. La distancia focal y la información de texturas disponible como resultado de la calidad de las imágenes capturadas, son elementos que se deben valorar en función de la calidad de los resultados.

Conclusiones generales

El estudio realizado permitió evaluar las tendencias actuales en que se realiza el proceso de detección de vitalidad y los diferentes enfoques propuestos en la bibliografía consultada. Se identificó que las investigaciones actuales están orientadas al estudio de tres propuestas: *análisis de las texturas*, *detección de movimientos* y *análisis de signos de vida*.

Esta investigación enfocada en el cumplimiento de los objetivos trazados quedó enmarcada en el *análisis de las texturas*, implementando la *detección de movimiento en la región de los ojos* y el *análisis de la variación del grado de enfoque*.

Como resultado del diseño se implementó un componente utilizando C# como lenguaje de programación, se definió una arquitectura en dos capas lo que permitió obtener una solución extensible, orientada a la posterior inclusión de nuevos mecanismos de detección de vitalidad.

La fase de implementación permitió obtener una versión funcional de la propuesta de solución. La combinación de los dos métodos implementados permite evaluar la ocurrencia de movimientos y la existencia de volumen en la imagen de rostro evaluada, ambos patrones determinan en gran medida la existencia frente a la cámara de un individuo con vida, dejando fuera la falsificación de rasgos faciales utilizando fotografías o videos.

Al concluir el proceso de desarrollo y realizadas las pruebas de rendimiento se determinó que la estrategia implementada para el análisis de textura basado en la evaluación del grado de enfoque, previene al sistema de reconocimiento facial de ataques en los cuales se haga uso de videos y fotos, garantizando un 90 % de efectividad en su aplicación. Además, la técnica de detección de movimiento implementada garantiza un 80 % de efectividad aportando seguridad a la metodología para lograr un sistema de identificación facial robusto ante ataques de suplantación de identidad.

Recomendaciones

Después de cumplido el objetivo general, teniendo implementado el componente que permite detectar vitalidad en las imágenes durante el proceso de captura, al concluir la investigación y luego de analizar los resultados obtenidos se recomienda lo siguiente:

1. Atendiendo a los métodos analizados en este trabajo, se debe realizar un estudio de factibilidad para su posterior implementación, priorizando aquellos que sean capaces de identificar fraudes que utilicen elementos que contengan las tres dimensiones. Tener en cuenta que los estudios basados en el *análisis de movimiento* y el *análisis de signos de vida* están propuestos como más promisorios, aunque estos enfoques requieren la colaboración del usuario y sus niveles de complejidad y costo económico aumentan, se debe considerar su inclusión como elemento detector de vitalidad.
2. Determinar una técnica más eficiente, podría aplicarse una máquina de soporte vectorial como estrategia para la evaluación y clasificación de los resultados, basados en el análisis del grado de vitalidad obtenido en cada método y los umbrales identificados en cada caso, de esa forma se pudiera determinar de manera consecuente la decisión con respecto al nivel vitalidad detectado.

Referencias bibliográficas

- [1] X. Tan, Y. Li, J. Liu, and L. Jiang, “*Face Liveness Detection from A Single Image with Sparse Low Rank Bilinear Discriminative Model*,” In *ECCV*, 2010.
- [2] B. Peixoto, C. Michelassi, and A. Rocha, “*Face liveness detection under bad illumination conditions*,” In *ICIP*, pp. 3557–3560, 2011.
- [3] *IBM Data breach statistics*. <http://www-935.ibm.com/services/us/en/it-services/security-services/data-breach/>.
- [4] *Cybersecurity: The new business priority*. <http://www.pwc.com/us/en/view/issue-15/cybersecurity-business-priority.jhtml>.
- [5] C. Asensio, “*La seguridad informática en evolución constante*,” *Datamation: la revista española de tecnología de la información para empresa*, no. 211, pp. 44–45, 2004.
- [6] J. Anil K., F. Patrick, and A. R. Arun, “*Handbook of Biometrics*,” vol. Springer, 2008.
- [7] D. Ngo, “*Vietnamese security firm: Your face is easy to fake*.” http://news.cnet.com/8301-17938_105-10110987-1.html, 2008.
- [8] S. Hemalatha and W. Amitabh, “*A Study of Liveness Detection in Face Biometric Systems*,” vol. 91, pp. 31–35, abril 2014.
- [9] K. Kollreider, H. Fronthaler, and J. Bigun, “*Evaluating Liveness by Face Images and the Structure Tensor*,” vol. Automatic Identification Advanced Technologies, Fourth IEEE Workshop on 17-18 OCT 2005, pp. 75–80, 2005.
- [10] K. Chander and S. Nitin, “*Fake Face Recognition using Fusion of Thermal Imaging and Skin Elasticity*,” vol. IJCSCi vol4, pp. 65–72, marzo 2013.
- [11] G. Kim, S. Eum, J. K. Suhr, D. Kim, K. R. Park, and J. Kim, “*Face liveness detection based on texture and frequency analyses*,” vol. 5th IAPR International Conference on Biometrics (ICB), New Delhi, India, pp. 67–72, marzo 2012.

- [12] T. Ojala and M. Pietikainen, “*Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns*,” *IEEE*, vol. Transactions on Pattern Analysis and Machine Intelligence, pp. 971–987, 2012.
- [13] J. Bai, T.-T. Ng, X. Gao, and Y.-Q. Shi, “*Is Physics-based Liveness Detection Truly Possible with a Single Image?*,” *IEEE*, vol. In Proceedings of IEEE International Symposium on Circuits and Systems, pp. 3425–3428, 2010.
- [14] A. K. Jain, K. Nandakumar, and A. Ross, “*Score normalization in multimodal biometric systems*,” *IEEE*, vol. Pattern recognition, pp. 2270–2285, 2005.
- [15] H. S. Choi, R. C. Kang, K. Choi, A. T. B. Jin, and J. Kim, Fake Fingerprint Detection using Multiple Static Features, vol. Optical Engineering. 2009.
- [16] C.-C. Chang and C.-J. Lin, “*Libsvm: A library for support vector machines*.” <http://www.csie.ntu.edu.tw/~cjlin/libsvm>, 2(3):27, 2011.
- [17] S. Eum, J. K. Suhr, and J. Kim, “*Face Recognizability Evaluation for ATM Applications with Exceptional Occlusion Handling*,” *IEEE*, vol. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 82–89, 2011.
- [18] H. K. Jee, S. U. Jung, and J. H. Yoo, “*Liveness detection for embedded face recognition system*,” *International Journal of Biological and Medical Sciences*, vol. 1(4), p. 235–238, 2006.
- [19] S. Kim, S. Yu, K. Kim, Y. Ban, and S. Lee, “*Face liveness detection using variable focusing*,” *2013 International Conference*, vol. Biometrics (ICB), p. 1–6, 2013.
- [20] P. Gang, S. Lin, W. Zhaohui, and L. Shihong, “*Eyeblink-based anti-spoofing face Recognition from webcamera*,” *IEEE*, no. 978-1-4244-1631-8/07, 2007.
- [21] S. Lin, P. Gang, W. Zhaohui, and L. Shihong, “*Blinking-Based Live Face Detection Using Conditional Random Fields*,” *Seoul, Korea, International Conference*, pp. 252–260, agosto 2004.
- [22] P. Gang, S. Lin, W. Zhaohui, and L. Shihong, “*Eyeblink based Anti-Spoofing in Face Recognition from a Generic Webcamera*,” *Recent Advances in Face Recognition*, 2007.

-
- [23] G. Pan, W. Zhaohui, and S. Lin, “*Liveness Detection for Face Recognition*,” *Recent Advances in Face Recognition*, Book edited by: Kresimir Delac, Mislav Grgic and Marian Stewart Bartlett, ISBN 978-953-7619-34-3, 2008.
- [24] J. Lafferty, A. McCallum, and F. Pereira, “*Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data*,” *ICML’01*, pp. 282–289, 2001.
- [25] L. Rabiner, “*A tutorial on hidden markov models and selected applications in speech recognition*,” *Proceedings of the IEEE*, pp. 77(2):257–286, 1989.
- [26] “Zju eyeblink database, <http://www.cs.zju.edu.cn/~gpan> o <http://www.stat.ucla.edu/~gpan>,”
- [27] J. Yang, Z. Lei, S. Liao, and S. Li, “*Face Liveness Detection with Component Dependent Descriptor*,” *Biometrics (ICB)*, p. 1–6, 2013 International Conference.
- [28] Z. Zhang, J. Yan, S. Liu, Z. Lei, D. Yi, and S. Z. Li, “*A face antispoofing database with diverse attacks*,” *ICB*, p. 26–31.IEEE, 2012.
- [29] J. Maatta, A. Hadid, and M. Pietikainen, “*Face spoofing detection from single images using microtexture analysis*,” *Biometrics (IJCB)*, 2011 International Joint Conference, pp. 1–7.IEEE, octubre 2011.
- [30] K. Kollreider, H. Fronthaler, M. I. Faraj, and J. Bigun, “*Real-time face detection and motion analysis with application in liveness assessment*,” *IEEE Transactions on Information Forensics and Security*, pp. 2(3–2):548–558, 2007.
- [31] K. Kollreider, H. Fronthaler, and J. Bigun, “*Verifying Liveness by Multiple Experts in Face Biometrics*,” *In IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1–6, 2008.
- [32] R. G. Figueroa, C. J. Solís, and A. A. Cabrera, “*METODOLOGÍAS TRADICIONALES VS. METODOLOGÍAS ÁGILES*,” *Universidad Técnica Particular de Loja*.

- [33] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mallor, K. Shwaber, and J. Sutherland, “*The Agile Manifesto*,” tech. rep., The Agile Alliance, 2001.
- [34] J. H. Canós, P. Letelier, and M. C. Penadés, “*Metodologías Ágiles en el Desarrollo de Software*,” *Universidad Politécnica de Valencia*.
- [35] K. Beck, “*Extreme Programming Explained. Embrace Change*,” Addison Wesley, 1999.
- [36] J. Rumbaugh, I. Jacobson, and G. Booch, “*El Lenguaje Unificado de Modelado. Manual de referencia*,” Addison Wesley, 2007.
- [37] P. López González, A. A. López González, and J. A. Lázaro Gallud, “*Herramientas CASE ¿Cómo incorporarlas con éxito a nuestra organización?*,” *Universidad de Castilla-La Mancha*, 2005.
- [38] *Microsoft Developer Network Online Documentation*. <https://msdn.microsoft.com/es-es/library/67ef8sbd.aspx>.
- [39] MSDN, “Lo más destacado de microsoft visual studio.” <https://msdn.microsoft.com/es-es/library/vstudio/dd547188%28v=vs.100%29.aspx>.
- [40] MSDN, “Novedades de microsoft visual studio 2012.” <https://msdn.microsoft.com/es-es/library/vstudio/bb386063%28v=vs.110%29.aspx>.
- [41] G. Bradski, “Opencv library,” *Dr. Dobb’s Journal of Software Tools*, 2000.
- [42] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd edition, vol. Wiley-Interscience Publication. 2001.
- [43] P. Viola and M. Jones, “*Rapid Object Detection using a Boosted Cascade of Simple Features*,” *In Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 511–518, 2001.
- [44] H. Wang, S. Z. Li, and Y. Wang, “*Face Recognition under Varying Lighting Condition Using Self Quotient Image*,” *In Proceedings of International Conference on Automatic Face and Gesture Recognition*, pp. 819–824, 2004.

- [45] S. Nayar and Y. Nakagawa, “*Shape from Focus*,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, pp. 824–831, agosto 2005.
- [46] R. C. González and P. Woods, *Digital Image Processing*. Addison Wesley, 2002.
- [47] F. Amo Alonso, M. Loic, A. Normand, and F. J. Segovia, “*El Proceso Unificado está centrado en la arquitectura*,” in *Introducción a la ingeniería del software. s.l.*, pp. 337–338, Delta Publicaciones, 2005.
- [48] I. Sommerville, *Software Engineering*. Harlow, England: Addison-Wesley, 9 ed., 2010.
- [49] I. Molyneaux, *The Art of Application Performance Testing: Help for Programmers and Quality Assurance*. O’Reilly Media, Inc., 1st ed., 2009.

Bibliografía consultada

- Ambler, Scott, *Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process*, John Wiley & Sons, 2002.
- Alonso Fernando Amo, A. Normand Martínez Loic, and Francisco Javier Pérez Segovia, *Introducción a la ingeniería del software*, Delta Publicaciones, 2005.
- Hunt, John, *Agile software construction.*, Springer, 2006.
- R. Padilla, C. F. F. Costa Filho, and M. G. F. Costa, *Evaluation of Haar Cascade Classifiers Designed for Face Detection*, World Academy of Science, Engineering and Technology, vol.6, 2012.
- Shin Shi, *Emgu CV Essentials*, Packt Publishing, 2013.
- Ioannis G. Stamelos, and Panagiotis Sfetsos, *Agile Software Development Quality Assurance*, Idea Group Inc (IGI), 2007.
- Ernest López. Teniente, *Diseño de sistemas software en UML.*, Univ. Politécnica. de Catalunya, 2004.

Anexo A: Historias de usuario

Tabla A.1: HU Detectar un rostro a partir de una imagen.

Historia de usuario	
Número: CDHU01	Usuario: Sergio Lecier Soler González
Nombre: Detectar un rostro a partir de una imagen.	
Prioridad de negocio: Alta	Riesgo de desarrollo: Alto
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Sergio Lecier Soler González	
Descripción: A partir de una imagen de entrada verificar si existe un un rostro y en caso positivo extraer la región de la imagen que lo contiene.	
Observaciones: Se utiliza como apoyo la librería EmguCV que implementa el mecanismo necesario para la detección de rostro.	
Requisitos funcionales: Software	
Asuntos pendientes: Ninguno	

Tabla A.2: HU Extraer la región de los ojos.

Historia de usuario	
Número: CDHU02	Usuario: Sergio Lecier Soler González
Nombre: Extraer la región de los ojos.	
Prioridad de negocio: Alta	Riesgo de desarrollo: Alto
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Sergio Lecier Soler González	
Descripción: Teniendo como imagen de entrada la región de un rostro válido se identifican las regiones que ocupan los dos ojos.	
Observaciones: Se utiliza como apoyo la librería EmguCV que implementa el mecanismo necesario para la detección del área que ocupan los ojos.	
Requisitos funcionales: Software	

Asuntos pendientes: Ninguno

Tabla A.3: HU Extraer la región de la nariz.

Historia de usuario	
Número: CDHU03	Usuario: Sergio Lecier Soler González
Nombre: Extraer la región de la nariz.	
Prioridad de negocio: Alta	Riesgo de desarrollo: Alto
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Sergio Lecier Soler González	
Descripción: Teniendo como imagen de entrada la región de un rostro válido se identifica las región que ocupa la nariz.	
Observaciones: Se utiliza como apoyo la librería EmguCV que implementa el mecanismo necesario para la detección del área que ocupa la nariz.	
Requisitos funcionales: Software	
Asuntos pendientes: Ninguno	

Tabla A.4: HU Extraer la región de la oreja.

Historia de usuario	
Número: CDHU04	Usuario: Sergio Lecier Soler González
Nombre: Extraer la región de la oreja.	
Prioridad de negocio: Alta	Riesgo de desarrollo: Alto
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Sergio Lecier Soler González	
Descripción: Teniendo como imagen de entrada la región de un rostro válido se identifican los puntos característicos y a partir de estos se estima la región que ocupa la oreja desde la vista frontal.	

Observaciones: Se utiliza como apoyo la librería EmguCV y ASMPLAT que devuelven la información necesaria con respecto a los puntos característicos.
Requisitos funcionales: Software
Asuntos pendientes: Ninguno

Tabla A.5: HU Realizar el proceso de binarizado sobre una imagen.

Historia de usuario	
Número: CDHU05	Usuario: Sergio Lecier Soler González
Nombre: Realizar el proceso de binarizado sobre una imagen.	
Prioridad de negocio: Alta	Riesgo de desarrollo: Alto
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Sergio Lecier Soler González	
Descripción: A partir de una imagen de entrada se evalúa la intensidad de cada píxel, dicha intensidad se compara con un umbral de intensidad y se le asigna al píxel en cuestión el color negro o blanco en función del resultado de la comparación.	
Observaciones: Ninguna	
Requisitos funcionales: Software	
Asuntos pendientes: Ninguno	

Tabla A.6: HU Aumentar el grado de enfoque a una imagen.

Historia de usuario	
Número: CDHU06	Usuario: Sergio Lecier Soler González
Nombre: Aumentar el grado de enfoque a una imagen.	
Prioridad de negocio: Alta	Riesgo de desarrollo: Alto
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Sergio Lecier Soler González	

Descripción: Utilizando la imagen de entrada como un mapa de bit, se realiza una multiplicación por el kernel de convolución necesario con el objetivo de aumentar de forma manual el grado de enfoque en cada píxel de dicha imagen.
Observaciones: Ninguna
Requisitos funcionales: Software
Asuntos pendientes: Ninguno

Tabla A.7: HU Calcular la Distancia de Hamming entre dos imágenes binarizadas.

Historia de usuario	
Número: CDHU07	Usuario: Sergio Lecier Soler González
Nombre: Calcular la Distancia de Hamming entre dos imágenes binarizadas.	
Prioridad de negocio: Alta	Riesgo de desarrollo: Alto
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Sergio Lecier Soler González	
Descripción: Utilizando dos imágenes previamente binarizadas se realiza la comparación píxel a píxel ejecutando el operador XOR sobre cada par, se cuentan los pares de píxeles que no son iguales y se asume como la distancia entre las dos imágenes.	
Observaciones: Las imágenes deben tener la mismas dimensiones de ancho por alto.	
Requisitos funcionales: Software	
Asuntos pendientes: Ninguno	

Tabla A.8: HU Calcular la Suma Modificada de Laplace a partir de una imagen.

Historia de usuario	
Número: CDHU08	Usuario: Sergio Lecier Soler González
Nombre: Calcular la Suma Modificada de Laplace a partir de una imagen.	
Prioridad de negocio: Alta	Riesgo de desarrollo: Alto
Puntos estimados: 1	Iteración asignada: 1

Programador responsable: Sergio Lecier Soler González
Descripción: Utilizando una imagen como parámetro de entrada se le determina el grado de enfoque utilizando el método de Laplace (SML), para cada píxel de la imagen se realiza el cálculo según las ecuaciones 2.3 y 2.4.
Observaciones: Ninguna
Requisitos funcionales: Software
Asuntos pendientes: Ninguno

Anexo B: Tarjetas CRC

Tabla B.1: Tarjeta CRC Detector de movimiento.

Detector de movimiento	
Responsabilidades	Colaboradores
Obtener LS utilizando ojo derecho	Detector de vitalidad, Herramientas para imágenes
Obtener LS utilizando ojo izquierdo	Detector de vitalidad, Herramientas para imágenes
Evaluación de vitalidad	Detector de vitalidad, Herramientas para imágenes

Tabla B.2: Tarjeta CRC Detector de volumen.

Detector de volumen	
Responsabilidades	Colaboradores
Obtener LS utilizando características	Detector de vitalidad
Evaluación de vitalidad	Detector de vitalidad

Tabla B.3: Tarjeta CRC Imagen de rostro.

Imagen de rostro	
Responsabilidades	Colaboradores
Obtener rostro	Verificador de imagen
Obtener oreja izquierda	Verificador de imagen
Obtener nariz	Verificador de imagen
Enfocar región de la nariz	Herramientas de imagen
Obtener ojos	Región de ojos

Anexo C: Tareas de ingeniería

Tabla C.1: Tareas de ingeniería de la primera iteración.

Iteración 1	
Historia de usuario	Tareas
Detectar un rostro a partir de una imagen.	<ol style="list-style-type: none">1. Recibir la imagen.2. Ejecutar el reconocimiento de patrones y detectar el rostro si existe.3. Definir los puntos característicos.
Extraer la región de los ojos.	<ol style="list-style-type: none">1. Obtener una imagen de entrada que contenga un rostro ya detectado.2. Ejecutar el reconocimiento de patrones y extraer las regiones de ambos ojos.
Extraer la región de la nariz.	<ol style="list-style-type: none">1. Obtener una imagen de entrada que contenga un rostro ya detectado.2. Ejecutar el reconocimiento de patrones y extraer las región de la nariz.

Extraer la región de la oreja.	<ol style="list-style-type: none">1. Obtener una imagen de entrada que contenga un rostro ya detectado.2. En función de los puntos característicos identificados en el rostro detectado, definir la región que contiene el área de la oreja izquierda desde una vista frontal.
Realizar el proceso de binarizado sobre una imagen.	<ol style="list-style-type: none">1. Obtener una imagen de entrada.2. Por cada píxel, calcular el nivel de intensidad.3. Comparar el nivel de intensidad y definir el color que se le asignará en función del resultado de la comparación.

<p>Aumentar el grado de enfoque a una imagen.</p>	<ol style="list-style-type: none">1. Obtener una imagen de entrada.2. Seleccionar el vector de convolución necesario.3. Definir la intensidad con que se ejecutará el filtro.4. Ejecutar el proceso de convolución utilizando la imagen y el vector seleccionado.
<p>Calcular la Distancia de Hamming entre dos imágenes binarizadas.</p>	<ol style="list-style-type: none">1. Obtener dos imágenes de entrada.2. Igualar la dimensión de las dos imágenes.3. Calcular la cantidad de píxeles que no son iguales.4. Definir la cantidad de píxeles diferentes como el valor de la distancia.

Calcular la Suma Modificada de Laplace a partir de una imagen.

1. Obtener un imagen de entrada.
2. Seleccionar la ecuación 2.4.
3. Seleccionar el paso a utilizar.
4. Por cada píxel evaluar la ecuación y sumar.

Anexo D: Resultado de las pruebas unitarias

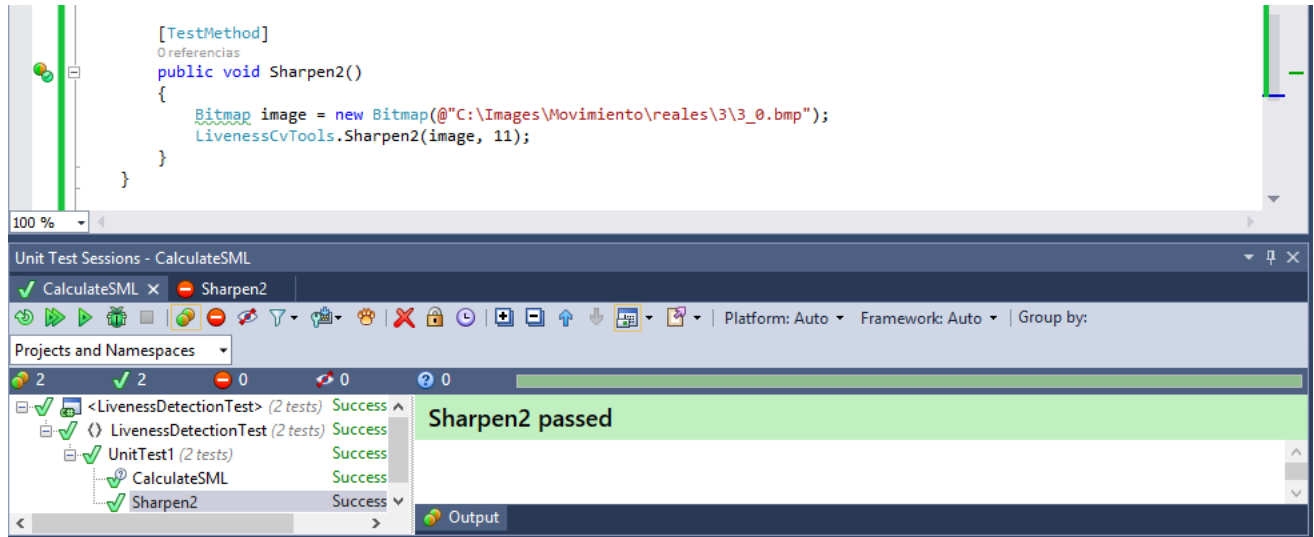


Figura D.1: Función para realizar el grado de enfoque.

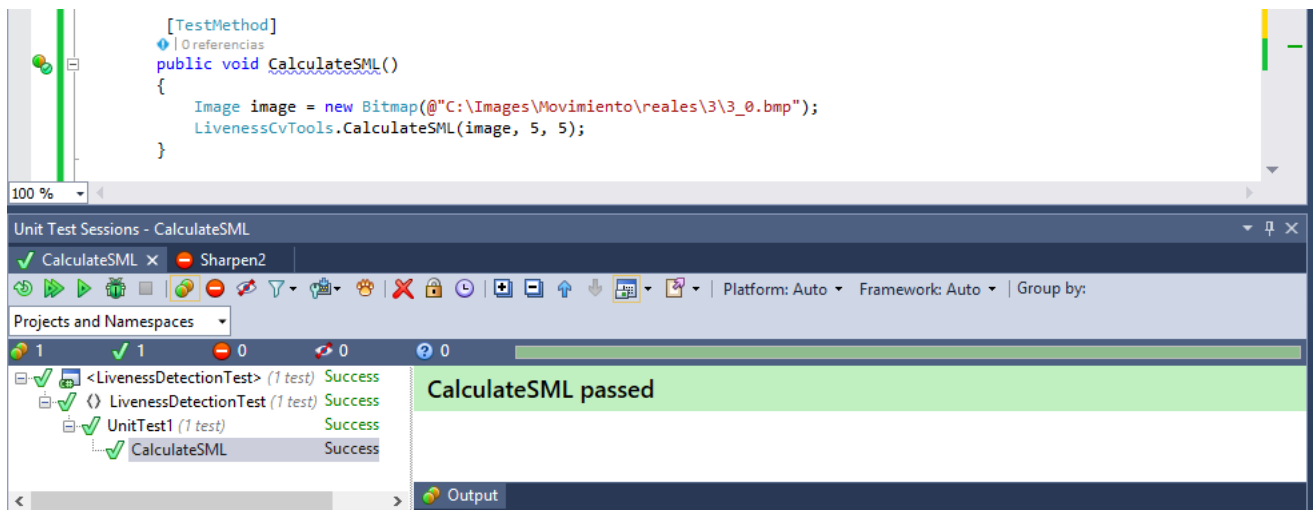


Figura D.2: Función para calcular el grado de enfoque.

Anexo E: Casos de prueba alfa

Tabla E.1: CP Detectar un rostro a partir de una imagen.

Caso de prueba de aceptación	
Código: CDHU01CP01	HU: Detectar un rostro a partir de una imagen
Responsable de la prueba: Sergio Lecier Soler González.	
Descripción: Prueba para verificar que dado una imagen que contiene un rostro el componente es capaz de extraer la región que lo contiene.	
Condiciones de ejecución: El componente debe recibir una imagen de rostro válido.	
Entrada/Pasos de ejecución: <ol style="list-style-type: none">1. Ejecutar la función pasándole la imagen que contiene el rostro.2. Obtener la sección de imagen que contiene el rostro.	
Resultado esperado: El sistema debe ser capaz de detectar la existencia de un rostro en la imagen y devolver dicha región.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla E.2: CP Extraer la región de los ojos.

Caso de prueba de aceptación	
Código: CDHU02CP02	HU: Extraer la región de los ojos
Responsable de la prueba: Sergio Lecier Soler González.	
Descripción: Prueba para verificar que dado una imagen que contiene un rostro el componente es capaz de extraer la región que contiene cada ojo.	
Condiciones de ejecución: El componente debe recibir una imagen de rostro válido.	

<p>Entrada/Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. Ejecutar la función pasándole la imagen que contiene el rostro. 2. Obtener las secciones de la imagen que contiene cada ojo.
<p>Resultado esperado: El sistema debe ser capaz de detectar la existencia de un rostro en la imagen y devolver cada región de los ojos.</p>
<p>Evaluación de la prueba: Prueba satisfactoria.</p>

Tabla E.3: CP Extraer la región de la nariz.

Caso de prueba de aceptación	
Código: CDHU03CP03	HU: Extraer la región de la nariz
Responsable de la prueba: Sergio Lecier Soler González.	
Descripción: Prueba para verificar que dado una imagen que contiene un rostro el componente es capaz de extraer la región que contiene la nariz.	
Condiciones de ejecución: El componente debe recibir una imagen de rostro válido.	
<p>Entrada/Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. Ejecutar la función pasándole la imagen que contiene el rostro. 2. Obtener la sección de la imagen que contiene la nariz. 	
Resultado esperado: El sistema debe ser capaz de detectar la existencia de un rostro en la imagen y devolver la región que contiene el área de la nariz.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla E.4: CP Extraer la región de la oreja.

Caso de prueba de aceptación

Código: CDHU04CP04	HU: Extraer la región de la oreja
Responsable de la prueba: Sergio Lecier Soler González.	
Descripción: Prueba para verificar que dado una imagen que contiene un rostro el componente es capaz de extraer la región que contiene la oreja desde una vista frontal.	
Condiciones de ejecución: El componente debe recibir una imagen de rostro válido.	
Entrada/Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Ejecutar la función pasándole la imagen que contiene el rostro. 2. Obtener la sección de la imagen que contiene la oreja. 	
Resultado esperado: El sistema debe ser capaz de detectar la existencia de un rostro en la imagen y devolver la región que contiene el área de la oreja.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla E.5: CP Realizar el proceso de binarizado sobre una imagen.

Caso de prueba de aceptación	
Código: CDHU05CP05	HU: Realizar el proceso de binarizado sobre una imagen
Responsable de la prueba: Sergio Lecier Soler González.	
Descripción: Prueba para verificar que dada una imagen el sistema sea capaz de realizar el proceso de binarizado evaluando la intensidad de cada píxel.	
Condiciones de ejecución: El componente debe recibir una imagen, en este caso se recomienda que sea la sección que contenga uno de los dos ojos.	
Entrada/Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Ejecutar la función pasándole la imagen. 2. Obtener la imagen nueva. 	
Resultado esperado: El sistema debe devolver la nueva imagen binaria.	

Evaluación de la prueba: Prueba satisfactoria.

Tabla E.6: CP Aumentar el grado de enfoque a una imagen.

Caso de prueba de aceptación	
Código: CDHU06CP06	HU: Aumentar el grado de enfoque a una imagen
Responsable de la prueba: Sergio Lecier Soler González.	
Descripción: Prueba para verificar que dada una imagen el sistema sea capaz de realizar el enfocado de forma manual.	
Condiciones de ejecución: El componente debe recibir una imagen, en este caso se recomienda que tenga poco grado de enfoque.	
Entrada/Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Ejecutar la función pasándole la imagen. 2. Obtener la imagen nueva. 	
Resultado esperado: El sistema debe devolver la nueva imagen enfocada.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla E.7: CP Calcular la Distancia de Hamming entre dos imágenes binarizadas.

Caso de prueba de aceptación	
Código: CDHU07CP07	HU: Calcular la Distancia de Hamming entre dos imágenes binarizadas
Responsable de la prueba: Sergio Lecier Soler González.	
Descripción: Prueba para verificar que dadas dos imágenes el sistema sea capaz de realizar la operación de comparación.	
Condiciones de ejecución: El componente debe recibir dos imágenes binarizadas.	

<p>Entrada/Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. Ejecutar la función pasándole las dos imágenes. 2. Obtener el valor.
<p>Resultado esperado: El sistema debe devolver la distancia, representando la diferencia de una imagen con respecto a la otra.</p>
<p>Evaluación de la prueba: Prueba satisfactoria.</p>

Tabla E.8: CP Calcular la Suma Modificada de Laplace a partir de una imagen.

Caso de prueba de aceptación	
Código: CDHU08CP08	HU: Calcular la Suma Modificada de Laplace a partir de una imagen
Responsable de la prueba: Sergio Lecier Soler González.	
Descripción: Prueba para verificar que dada una imagen el sistema sea capaz de calcular la Suma Modificada de Laplace con respecto al grado de enfoque que presenta la misma.	
Condiciones de ejecución: El componente debe recibir una imagen.	
<p>Entrada/Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. Ejecutar la función pasándole una imagen. 2. Obtener el valor. 	
Resultado esperado: El sistema debe devolver el valor de la suma, representando el grado de enfoque de dicha imagen.	
Evaluación de la prueba: Prueba satisfactoria.	