



---

**Universidad de Valladolid**

TRABAJO DE FIN DE GRADO

---

**Reconocimiento parcial de  
huellas dactilares mediante Redes  
Neuronales Artificiales**

---

*Autor:*

Mario Rodríguez García

*Tutor:*

Teodoro Calonge Cano

20 de Enero de 2017



# Contenidos

<b>1</b>	<b>Introducción</b>	<b>11</b>
<b>2</b>	<b>Modelado y diseño del sistema</b>	<b>15</b>
2.1	Casos de uso . . . . .	16
2.1.1	Descripción de los casos de uso . . . . .	17
2.2	Diagrama de clases . . . . .	31
2.3	Diagrama entidad relación . . . . .	39
2.4	Diagrama relacional . . . . .	39
<b>3</b>	<b>Binarización</b>	<b>41</b>
3.1	Máscara de filtro . . . . .	41
3.2	Imagen de orientaciones . . . . .	45
3.3	Procesamiento píxel a píxel . . . . .	47
<b>4</b>	<b>Adelgazamiento</b>	<b>51</b>
4.1	Algoritmo Guo-Hall . . . . .	52
<b>5</b>	<b>Extracción de minutias</b>	<b>55</b>
5.1	Red Neuronal: Perceptrón Multicapa . . . . .	55
5.2	Entrenamiento de la red neuronal . . . . .	59
5.3	Extracción de minutias píxel a píxel . . . . .	60
<b>6</b>	<b>Comparación de dos huellas dactilares</b>	<b>61</b>
6.1	Búsqueda del par principal . . . . .	62
6.2	Búsqueda de los pares coincidentes . . . . .	64
6.3	Verificación de las huellas dactilares . . . . .	67
<b>7</b>	<b>Conclusiones</b>	<b>69</b>
7.1	Trabajo futuro . . . . .	70
<b>8</b>	<b>Anexo I: manual de instalación</b>	<b>73</b>

<b>9 Anexo II: manual de usuario</b>	<b>75</b>
9.1 Verificación de huellas dactilares . . . . .	76
9.2 Identificación de un individuo . . . . .	79
9.3 Gestión de individuos . . . . .	80
9.4 Creación de un individuo . . . . .	81
9.5 Ver detalles de individuo . . . . .	83
9.6 Comprobar eficiencia de un perceptrón . . . . .	85
9.7 Obtener minutiaes y minutiaes no repetidos . . . . .	86
<b>10 Anexo III: manual de despliegue en <i>Visual Studio Community</i> 2015</b>	<b>89</b>

# Lista de figuras

1.1	Ejemplos de huellas de mala calidad. . . . .	12
1.2	Tipos de minutiaes en función de su forma . . . . .	13
1.3	Diagrama de actividades de un sistema de autenticación de huellas. 14	
2.1	Diagrama de los casos de uso de la aplicación. . . . .	16
2.2	Diagrama de clases de diseño del sistema desarrollado. . . . .	31
2.3	Diagrama entidad relación de la base de datos. . . . .	39
2.4	Diagrama relacional de la base de datos. . . . .	39
3.1	Imagen en 3D de los valores del filtro generado. . . . .	44
3.2	Ejemplo del proceso de extracción de la imagen de orientaciones de una huella dactilar. . . . .	46
3.3	Imagen del resultado del suavizado en la imagen de orientaciones de la huella de la figura 3.2a. . . . .	47
3.4	Comparación entre la imagen original y la binarizada. . . . .	48
3.5	Resultado del suavizado sobre la imagen binarizada de la figura 3.4b. . . . .	49
4.1	Resultado del adelgazamiento sobre la imagen binarizada de la figura 3.5. . . . .	53
5.1	Arquitectura de un perceptrón multicapa con dos capas ocultas. 56	
5.2	Primer diseño de la red neuronal para la extracción de minutiaes. 56	
5.3	Red neuronal con especialización. . . . .	57
5.4	Red neuronal con especialización y sistema de votación. . . . .	58
5.5	Diferentes ejemplos de patrones de bifurcaciones (primera fila), finales de línea (segunda fila) y no minutiaes (tercera fila). . . . .	59
5.6	Ejemplo de huella dactilar binarizada y adelgazada con los minutiaes resaltados. . . . .	60
6.1	Ejemplo de la conversión de bifurcaciones en fines de línea en la fase de binarización. . . . .	61
8.1	Instalador de la aplicación. . . . .	73
8.2	Instalación de Visual C++. . . . .	74

9.1	Vista principal de la aplicación . . . . .	75
9.2	Vista principal de la verificación de huellas. . . . .	76
9.3	Vista al seleccionar la huella dactilar . . . . .	76
9.4	Vista al obtener una huella dactilar desde el escáner. . . . .	77
9.6	Vista para guardar la huella dactilar y sus imágenes. . . . .	78
9.7	Vista del resultado de guardar las imágenes en una carpeta. . . . .	78
9.8	Vista del resultado de la verificación de dos huellas dactilares. . . . .	79
9.9	Vista principal de la identificación de un individuo mediante su huella dactilar. . . . .	79
9.10	Vista del resultado de la identificación de un individuo. . . . .	80
9.11	Vista principal de la gestión de individuos . . . . .	80
9.12	Vista principal de la creación de un individuo. . . . .	81
9.13	Vista de la captura de huellas dactilares para un individuo. . . . .	82
9.14	Vista del resultado de capturar una huella dactilar para un individuo. . . . .	82
9.15	Vista principal de los detalles de un individuo, así como de sus huellas dactilares. . . . .	83
9.16	Ejemplo de eliminación de una huella dactilar. . . . .	84
9.17	Ejemplo de eliminación de un individuo. . . . .	84
9.18	Vista principal del método Hold-Out para el perceptrón. . . . .	85
9.19	Vista de selección del conjunto de datos para el aprendizaje y prueba del perceptrón. . . . .	85
9.20	Ejemplo de ventana con todos los parámetros ajustados. . . . .	86
9.21	Vista del resultado del método Hold-Out. . . . .	86
9.22	Vista de la selección de la carpeta para la extracción de minutias de una huella. . . . .	87
9.23	Vista del resultado de la extracción de minutias sobre una huella. . . . .	87

# Resumen

El objetivo de este trabajo es la investigación, modelado y desarrollo de un sistema de identificación automática de huellas dactilares. Las personas pueden ser identificadas por los puntos característicos ubicados en las huellas dactilares, denominados minutias. Estos son extraídos a partir de su imagen en escala de grises, haciendo uso de una Red Neuronal Artificial. Con el fin de mejorar la eficiencia de la red neuronal, se realiza un preprocesado, consistente en la binarización de la imagen, es decir, la conversión de los grises en negro y blanco únicamente, según corresponda, y el adelgazamiento de las líneas a un solo píxel de grosor. Una vez se ha obtenido el conjunto de minutias, deben ser comparados con otro conjunto, mediante un algoritmo de comparación de patrones bidimensionales que determinará si ambas huellas pertenecen a la misma persona o no.





# Abstract

The goal of this work is research, modeling and develop an automated fingerprint identification system(AFIS). Fingerprints have a set of unique characteristic points, each one called minutiae. Fingerprint minutiaes have a different 2D pattern for each person. It can be extracted from a fingerprint grayscale image with the help of an artificial neural network. In order to improve the performance of that artificial neural network, it is highly recommended to carry out a preprocess. Basically, it consists on two steps: the binarization process, that converts a grayscale image to a black and white one, and the skeletonization task, which reduces the lines wide to only one pixel. After the minutiae set extraction, it must be contrasted with the other ones. To do that, a 2D pattern comparison algorithm is used, obtaining the fingerprint that corresponds to a given set of minutiaes.



# Capítulo 1

## Introducción

La piel de los dedos de los seres humanos está formada por conjuntos de crestas y valles que se originan durante el proceso de gestación y conforman la huella dactilar de una persona. Todas ellas son únicas, lo que implica que ningún ser humano tiene alguna de ellas idénticas entre sí, ni como las de otro ser humano, aunque, en teoría, dos personas con al menos una huella dactilar idéntica solo coexisten una vez cada  $10^{48}$  años. En la práctica, solo el 0,01 % de ellas son asociadas por error a otro individuo, lo cual es debido, en la mayoría de los casos, a la falta de calidad de la muestra obtenida para comparar.

La unicidad de las huellas dactilares permite que cualquier individuo esté inequívocamente identificado mediante cualquiera de ellas. Su uso como identificador único de una persona es una práctica muy extendida en las ciencias forenses desde hace muchos años; sin embargo, esta técnica no es empleada apenas en el resto de ámbitos en los que es necesario distinguir a un individuo de otro. La determinación de la identidad de una persona, en la mayoría de sitios, se sigue realizando con la ayuda de medios más tradicionales, como las llaves, las tarjetas identificativas o las contraseñas. Aunque la detección de un sujeto mediante huellas dactilares no está aún muy extendida en la sociedad, el abaratamiento de los dispositivos para su obtención en la última década está facilitando la introducción de este método en cada vez más ámbitos; algunos ejemplos de ello son los dispositivos móviles o los ordenadores portátiles, que ya vienen con un lector de huellas dactilares incorporado en muchos de sus modelos.

Las huellas dactilares representan un método identificativo más seguro que los tradicionales, que a menudo son sencillos de robar, suplantar u obtener. Pese a que la seguridad sea una de sus grandes ventajas, hasta hace poco requería de un equipo de personas entrenadas y formadas para el reconocimiento de huellas dactilares, lo que lo convertía en una opción inviable para casi todos los escenarios en los que se necesitara determinar la identidad de un individuo. Tras la llegada de la informática y el desarrollo de las Redes Neuronales Artificiales, el proceso de identificación de un individuo se vio reducido a la ejecución de

un programa que en escasos minutos conseguía un resultado, en muchas de las ocasiones, mejor que el del experto humano.

El proceso de reconocimiento de una huella dactilar comienza por su obtención para su posterior análisis. Existen dos métodos para conseguirlas: la impresión con tinta negra en un fragmento de papel o el uso de un escáner digital. Hay varios factores que han de ser tenidos en cuenta a la hora de su adquisición:

- Humedad del dedo.
- Presión aplicada al papel o al escáner con el dedo.

Ambos parámetros influyen sustancialmente en la calidad de la huella dactilar obtenida, como puede ser apreciado en la siguiente figura.



Figura 1.1: Ejemplos de huellas de mala calidad.

La siguiente fase del AFIS es la localización de los minutiaes, los cuales son puntos presentes en las huellas, que son usados para compararlos con los de otras, y de esta manera, determinar si ambas pertenecen al mismo individuo o no. Existen diferentes tipos en función de su forma, como ilustra la figura siguiente.



Figura 1.2: Tipos de minuciaes en función de su forma

Tras haber obtenido todos los minuciaes, éstos son comparados con los de otra huella dactilar uno por uno, hasta que se emparejan todos los posibles de la primera huella con los de la segunda. En función del porcentaje de minuciaes emparejados, se puede determinar si las huellas son del mismo dedo, o no, y por lo tanto si pertenecen, o no, al mismo individuo.

En este trabajo de fin de grado, se ha modelado e implementado un sistema automático de identificación de huellas dactilares (AFIS). Este método tiene los mismos pasos que el proceso tradicional: la extracción de minuciaes y su comparación entre huellas. El primer paso se consigue mediante el uso de una Red Neuronal Artificial y, el segundo, a través de un algoritmo de comparación de patrones bidimensionales.

El proceso de extracción de minuciaes mediante una Red Neuronal Artificial es posible realizarlo directamente sobre la huella dactilar obtenida por el escáner,

o la obtenida mediante el uso de papel y tinta. Sin embargo, la eficiencia de la Red Neuronal Artificial no sería óptima y, en algunos casos, no encontraría los minutiaes correctamente, o peor aún, encontraría algunos espurios. Con el fin de mejorar la eficiencia de la Red Neuronal Artificial, se elimina la máxima cantidad de información innecesaria de la imagen. Esto se consigue realizando un preprocesado, consistente en su binarización, que convierte la huella original en escala de grises. Posteriormente, pasaría a una imagen en blanco y negro exclusivamente, para su adelgazamiento, y así obtener líneas de un píxel de grosor.

El proceso completo de un AFIS es representado mediante el diagrama de actividades de la figura 1.3 de la página 14.

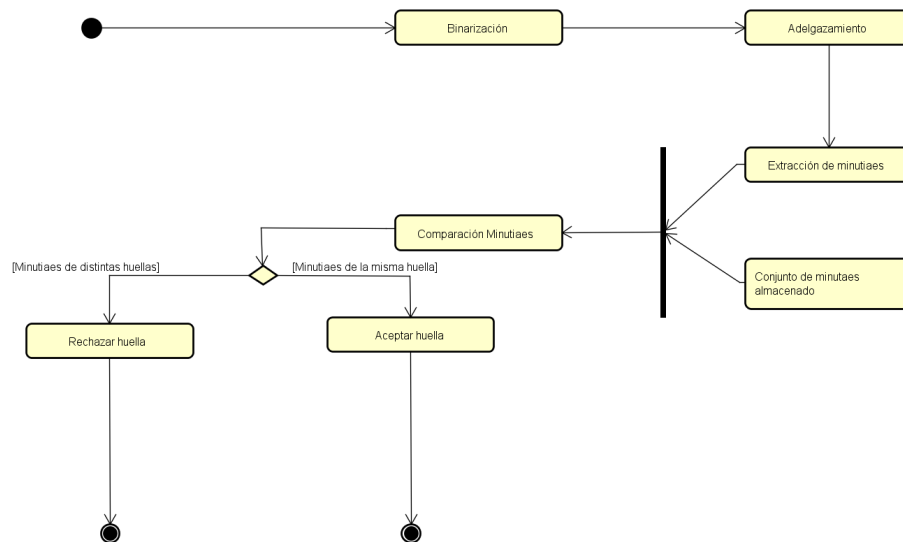


Figura 1.3: Diagrama de actividades de un sistema de autenticación de huellas.

La implementación de este trabajo se ha realizado en el lenguaje de programación Visual C++ con el entorno de Visual Studio 2015 y el gestor de bases de datos SQLite. Como complemento al lenguaje, se han usado las librerías: Qt para el desarrollo de la interfaz gráfica, CImg y las asociadas a los formatos TIFF, PNG y JPG para el tratamiento de imágenes, y, OpenMP para la paralelización de fragmentos de código. El motivo de la elección de este lenguaje es su buena eficiencia y que es orientado a objetos. Todas las librerías han sido escogidas debido a que estaban bien documentadas, facilitando así su uso, y a su sencilla integración en Visual Studio y Visual C++. En el caso de OpenMP, aparte de estas características, también se eligió porque su uso y funcionamiento ha sido explicado en la asignatura de Computación Paralela.



## Capítulo 2

# Modelado y diseño del sistema

### 2.1 Casos de uso





## 2.1.1 Descripción de los casos de uso

CU-001	Verificar huellas dactilares	
Versión	2.0 (15/06/2016)	
Autores	Mario Rodríguez García	
Descripción	El sistema deberá permitir al actor usuario comprobar si dos huellas pertenecen al mismo individuo según se describe en el siguiente caso de uso.	
Precondición	El usuario debe estar en la pantalla del menú principal.	
Secuencia normal	Paso	Acción
	1	El actor usuario selecciona la opción de <i>Verificar Huellas Dactilares</i> en la ventana principal.
	2	El sistema muestra la ventana de verificación de huellas dactilares.
	3	El actor usuario selecciona la opción de "Cargar Huella Dactilar" o la de "Obtener Huella del Escáner" de una de las dos posibles huellas dactilares.
	4	El sistema ejecuta el caso de uso <i>CU-0003</i> o <i>CU-004</i> , en función de la elección del usuario.
	5	El actor usuario selecciona la opción de <i>Cargar Huella Dactilar</i> o la de <i>Obtener Huella del Escáner</i> de la huella dactilar restante.
	6	El sistema ejecuta el caso de uso <i>CU-0003</i> o <i>CU-004</i> , en función de la elección del usuario.
	7	El actor usuario selecciona la opción de <i>Comparar Huellas Dactilares</i> .
	8	El sistema muestra una ventana con las imágenes de las dos huellas dactilares con los minutiaes emparejados en ambas huellas y un texto informando de si ambas huellas pertenecen al mismo individuo.
Excepciones	Paso	Acción
	1, 3, 5, 7	El actor usuario cancela la operación. El caso de uso queda sin efecto.
	4	El caso de uso ejecutado queda sin efecto. Este caso de uso vuelve al paso 3.
	6	El caso de uso ejecutado queda sin efecto. Este caso de uso vuelve al paso 5.

CU-002	Identificar individuo	
Versión	2.0 (15/06/2016)	
Autores	Mario Rodríguez García	
Descripción	El sistema deberá permitir al actor usuario identificar a un individuo a través de una huella dactilar, según se describe en el siguiente caso de uso.	
Precondición	El usuario debe estar en la pantalla del menú principal.	
Secuencia normal	Paso	Acción
	1	El actor usuario selecciona la opción de <i>Identificar Individuo</i> en la ventana principal.
	2	El sistema muestra la ventana de identificación de individuo.
	3	El actor usuario selecciona la opción de <i>Cargar Huella Dactilar</i> o la de <i>Obtener Huella del Escáner</i> .
	4	El sistema ejecuta el caso de uso <i>CU-0003</i> o <i>CU-004</i> , en función de la elección del usuario.
	5	El actor usuario selecciona el botón <i>Identificar Usuario</i> .
Excepciones	6	El sistema compara la huella dactilar cargada con todas las almacenadas en la base de datos, hasta encontrar el individuo al que pertenece, muestra por pantalla la huella coincidente y también una ventana con los datos del individuo.
	Paso	Acción
	1, 3, 5	El actor usuario cancela la operación. El caso de uso queda sin efecto.
	4	El caso de uso ejecutado queda sin efecto. Este caso de uso vuelve al paso 3.
	6	El sistema no encuentra ningún individuo en la base de datos al que pertenezca la huella cargada. El sistema muestra una ventana mostrando el error y el caso de uso queda sin efecto.

CU-003	Cargar huella desde escáner	
Versión	2.0 (15/06/2016)	
Autores	Mario Rodríguez García	
Descripción	El sistema deberá permitir al actor usuario obtener una imagen de una huella dactilar con los minutiaes resaltados, a partir de una en escala de grises obtenida desde el escáner según se describe en el siguiente caso de uso.	
Precondición	El usuario debe estar en la pantalla de creación de verificación de huellas dactilares, identificación de un individuo o en los detalles de un individuo.	
Secuencia normal	Paso	Acción
	1	El sistema muestra por pantalla la imagen actual recibida desde el escáner.
	2	El actor usuario selecciona el botón <i>Capturar Huella Dactilar</i> .
	3	El sistema: binariza, adelgaza, extrae los minutiaes de la huella dactilar y los resalta con un cuadrado, rojo en el caso de las bifurcaciones, y verde en caso de los fines de línea.
Excepciones	Paso	Acción
	2	El actor usuario cancela la operación. El caso de uso queda sin efecto.

CU-004	Cargar huella dactilar desde imagen	
Versión	2.0 (15/06/2016)	
Autores	Mario Rodríguez García	
Descripción	El sistema deberá permitir al actor usuario obtener una imagen de una huella dactilar con los minutiaes resaltados, a partir de una en escala de grises según se describe en el siguiente caso de uso.	
Precondición	El usuario debe estar en la pantalla de creación de verificación de huellas dactilares, identificación de un individuo o en los detalles de un individuo.	
Secuencia normal	Paso	Acción
	1	El sistema muestra por pantalla una ventana de selección de archivos.
	2	El actor usuario selecciona la huella que desea cargar.
	3	El sistema binariza, adelgaza, extrae los minutiaes de la huella dactilar y los resalta con un cuadrado, rojo en el caso de las bifurcaciones, y verde en caso de los fines de línea.
Excepciones	Paso	Acción
	2	El actor usuario cancela la operación. El caso de uso queda sin efecto.

CU-005	Ver individuos en el sistema	
Versión	2.0 (15/06/2016)	
Autores	Mario Rodríguez García	
Descripción	El sistema deberá permitir al actor usuario ver la lista de los individuos registrados en la base de datos según se describe en el siguiente caso de uso.	
Precondición	El usuario debe estar en la pantalla del menú principal.	
Secuencia normal	Paso	Acción
	1	El actor usuario selecciona el botón <i>Gestión de Individuos</i> .
Excepciones	2	El sistema muestra en una nueva ventana la lista de individuos registrados en la base de datos con su NIF, nombre y apellidos.
	Paso	Acción
	1	El actor usuario cancela la operación. El caso de uso queda sin efecto.

CU-006	Comprobar eficiencia de un perceptrón	
Versión	2.0 (15/06/2016)	
Autores	Mario Rodríguez García	
Descripción	El sistema deberá permitir al actor usuario obtener la eficiencia de un perceptrón con un set de datos y unos parámetros determinados según se describe en el siguiente caso de uso.	
Precondición	El usuario debe estar en la pantalla del menú principal.	
Secuencia normal	Paso	Acción
	1	El actor usuario selecciona el botón <i>Comprobar Eficiencia Perceptrón</i> .
	2	El sistema muestra una ventana con los datos a rellenar por el actor usuario.
	3	El actor usuario introduce la ruta, o la selecciona mediante un dialogo de selección de archivos, del set de datos deseado.
	4	El actor usuario introduce el número de neuronas en cada capa separadas por una coma.
	5	El actor usuario introduce el parámetro A de la función de activación.
	6	El actor usuario introduce el parámetro B de la función de activación.
	7	El actor usuario introduce el momentum del perceptrón.
	8	El actor usuario introduce el ratio de aprendizaje del perceptrón.
	9	El actor usuario introduce el número de épocas del experimento.
	10	El actor usuario introduce el porcentaje de aprendizaje del experimento Hold-Out.
	11	El sistema ejecuta un experimento Hold-Out con los datos introducidos por pantalla y muestra la matriz de confusión resultante en un cuadro de diálogo nuevo.
Excepciones	Paso	Acción
	1, 3, 4, 5, 6, 7, 8, 9, 10	El actor usuario cancela la operación. El caso de uso queda sin efecto.
	3	El actor usuario introduce una ruta que no pertenece a un set de datos, o a un set de datos inconsistente. El sistema muestra un mensaje de error y el caso de uso queda sin efecto.
	4	El actor usuario introduce un formato erróneo en la línea de neuronas. El sistema muestra un mensaje de error y el caso de uso queda sin efecto.

CU-007	Extraer todos los puntos de una huella	
Versión	2.0 (15/06/2016)	
Autores	Mario Rodríguez García	
Descripción	El sistema deberá permitir al actor usuario obtener todos los puntos de una huella dactilar procesada según se describe en el siguiente caso de uso.	
Precondición	El usuario debe estar en la pantalla del menú principal.	
Secuencia normal	Paso	Acción
	1	El actor usuario selecciona el botón <i>Extraer Minutias de una Huella</i> .
	2	El sistema muestra un cuadro de diálogo para seleccionar archivos.
	3	El actor usuario selecciona la carpeta donde se encuentra la huella procesada.
	4	El sistema extrae todos los puntos de la huella a un archivo denominado <i>dataset.csv</i> .
Excepciones	Paso	Acción
	1, 3	El actor usuario cancela la operación. El caso de uso queda sin efecto.
	4	El actor usuario selecciona una carpeta donde no existe una huella procesada. El sistema muestra un mensaje de error y caso de uso queda sin efecto.

CU-008	Extraer todos los puntos sin repetir de una huella	
Versión	2.0 (15/06/2016)	
Autores	Mario Rodríguez García	
Descripción	El sistema deberá permitir al actor usuario obtener todos los puntos sin repetir de varias huellas dactilares procesada según se describe en el siguiente caso de uso.	
Precondición	El usuario debe estar en la pantalla del menú principal.	
Secuencia normal	Paso	Acción
	1	El actor usuario selecciona el botón <i>Extraer Minutias Sin Repetir de una Huella</i> .
	2	El sistema muestra un cuadro de diálogo para seleccionar archivos.
	3	El actor usuario selecciona la carpeta donde se encuentran las carpetas con las huellas procesadas.
	4	El sistema extrae todos los puntos sin repetir de todas las huellas a un archivo denominado <i>dataset.csv</i> .
Excepciones	Paso	Acción
	1, 3	El actor usuario cancela la operación. El caso de uso queda sin efecto.
	4	El actor usuario selecciona una carpeta donde alguna o todas las carpetas no contienen huellas procesadas. El caso de uso queda continúa omitiendo la carpeta o carpetas.

CU-009	Añadir un individuo al sistema	
Versión	2.0 (15/06/2016)	
Autores	Mario Rodríguez García	
Descripción	El sistema deberá permitir al actor usuario agregar un individuo a la base de datos del sistema según se describe en el siguiente caso de uso.	
Precondición	El usuario debe estar en la pantalla de gestión de individuos.	
Secuencia normal	Paso	Acción
	1	El actor usuario selecciona el botón de <i>Añadir Nuevo Individuo</i> en la pantalla de gestión de individuos.
	2	El sistema muestra una nueva ventana con la pantalla de creación de individuos.
	3	El actor usuario introduce el NIF, nombre y apellidos del nuevo individuo
	4	El actor usuario selecciona el botón <i>Seleccionar Huellas Dactilares del Individuo</i> .
	5	El sistema ejecuta el caso de uso <i>CU-012</i> .
	6	El actor usuario selecciona el botón <i>Obtener Huellas Dactilares del escáner</i> .
	7	El sistema ejecuta el caso de uso <i>CU-011</i> .
	8	El actor usuario selecciona el botón <i>Crear Individuo</i> .
	9	El sistema procesa las huellas dactilares y almacena el individuo junto con sus huellas en la base de datos.
Excepciones	Paso	Acción
	1, 3, 4, 6, 8	El actor usuario cancela la operación. El caso de uso queda sin efecto.
	4	El actor usuario no carga las huellas dactilares desde una imagen o no carga ninguna huella dactilar. El caso de uso continúa por el paso 6.
	6	El actor usuario no carga las huellas dactilares desde el escáner o no carga ninguna huella dactilar. El caso de uso continúa por el paso 8.
	8	El actor usuario quiere cargar más huellas dactilares. El caso de uso continúa por el paso 4.
	8	El actor usuario no ha introducido ningún nombre, apellido, NIF o ha introducido un NIF inválido. El sistema muestra un error y el caso de uso continúa por el paso 3.
	8	El actor usuario ha seleccionado alguna huella perteneciente a otro individuo. El sistema no añade esa huella al nuevo individuo, muestra una advertencia y el caso de uso continúa.



CU-010	Eliminar un Individuo del Sistema	
Versión	2.0 (15/06/2016)	
Autores	Mario Rodríguez García	
Descripción	El sistema deberá permitir al actor usuario eliminar un individuo de la base de datos del sistema según se describe en el siguiente caso de uso.	
Precondición	El usuario debe estar en la pantalla de detalles de individuo.	
Secuencia normal	Paso	Acción
	1	El actor usuario selecciona el botón <i>Eliminar Individuo</i> en la pantalla de detalles de individuo.
	2	El sistema pide una confirmación de la acción.
	3	El actor usuario confirma la acción.
	4	El sistema elimina el individuo y todos sus datos asociados de la base de datos.
Excepciones	Paso	Acción
	1, 3	El actor usuario cancela la operación. El caso de uso queda sin efecto.

CU-011	Añadir una huella dactilar a un individuo desde escáner	
Versión	2.0 (15/06/2016)	
Autores	Mario Rodríguez García	
Descripción	El sistema deberá permitir al actor usuario añadir una huella dactilar a un individuo según se describe en el siguiente caso de uso.	
Precondición	El usuario debe estar en la pantalla de detalles de individuo o creación de individuos.	
Secuencia normal	Paso	Acción
	1	El actor usuario selecciona el botón <i>Añadir Huella desde Escáner</i> .
	2	El sistema muestra en una nueva ventana la imagen actual recibida desde el escáner.
	3	El actor usuario selecciona el botón <i>Guardar Huella Dactilar</i> .
	4	El sistema binariza, adelgaza, extrae los minutiaes de la huella dactilar y la almacena en la base de datos.
Excepciones	Paso	Acción
	1, 3	El actor usuario cancela la operación. El caso de uso queda sin efecto.
	4	El actor usuario ha seleccionado alguna huella perteneciente a otro individuo. El sistema no añade esa huella al nuevo individuo, muestra una advertencia y el caso de uso continúa.

CU-012	Añadir una huella dactilar a un individuo desde imagen	
Versión	2.0 (15/06/2016)	
Autores	Mario Rodríguez García	
Descripción	El sistema deberá permitir al actor usuario añadir una huella o huellas desde sus respectivas imágenes según se describe en el siguiente caso de uso.	
Precondición	El usuario debe estar en la pantalla de detalles de individuo o creación de individuos.	
Secuencia normal	Paso	Acción
	1	El actor usuario selecciona el botón <i>Añadir Huella desde Imagen</i> .
	2	El sistema muestra un selector de ficheros.
	3	El actor usuario selecciona en el selector de ficheros la huella o huellas que desea añadir al individuo.
	4	El sistema binariza, adelgaza, extrae los minutiae de la huella o huellas y las almacena en la base de datos.
Excepciones	Paso	Acción
	1, 3	El actor usuario cancela la operación. El caso de uso queda sin efecto.
	4	El actor usuario ha seleccionado alguna huella perteneciente a otro individuo. El sistema no añade esa huella al nuevo individuo, muestra una advertencia y el caso de uso continúa.

CU-013	Almacenar huella dactilar en PC	
Versión	2.0 (15/06/2016)	
Autores	Mario Rodríguez García	
Descripción	El sistema deberá permitir al actor usuario almacenar las diversas vistas de una huella en el ordenador según se describe en el siguiente caso de uso.	
Precondición	El usuario debe estar en la pantalla de creación de verificación de huellas dactilares, identificación de un individuo o en los detalles de un individuo.	
Secuencia normal	Paso	Acción
	1	El actor selecciona el botón <i>Guardar en PC</i> en la vista de detalles de individuo, verificación de huellas o identificación de individuo.
	2	El sistema muestra un selector de ficheros.
	3	El actor usuario selecciona la ruta donde desea almacenar las imágenes de la huella dactilar.
	4	El sistema guarda en esa ruta las diversas imágenes de la huella dactilar.
Excepciones	Paso	Acción
	1, 3	El actor usuario cancela la operación. El caso de uso queda sin efecto.

CU-014	Ver detalles de una huella dactilar de un individuo	
Versión	2.0 (15/06/2016)	
Autores	Mario Rodríguez García	
Descripción	El sistema deberá permitir al actor usuario ver los detalles de una huella dactilar según se describe en el siguiente caso de uso.	
Precondición	El usuario debe estar en la pantalla de detalles de un individuo.	
Secuencia normal	Paso	Acción
	1	El actor usuario hace clic en la huella deseada en la vista de detalles de individuo.
	2	El sistema carga la huella y la muestra por pantalla.
Excepciones	Paso	Acción
	1	El actor usuario cancela la operación. El caso de uso queda sin efecto.

CU-015	Eliminar una huella dactilar de un individuo	
Versión	2.0 (15/06/2016)	
Autores	Mario Rodríguez García	
Descripción	El sistema deberá permitir al actor usuario eliminar una huella dactilar de un individuo según se describe en el siguiente caso de uso.	
Precondición	El usuario debe estar en la pantalla de detalles de un individuo.	
Secuencia normal	Paso	Acción
	1	El actor usuario selecciona la huella dactilar deseada y pulsa suprimir en la vista de detalles de individuo.
	2	El sistema pide una confirmación de la acción.
	3	El actor usuario confirma la acción.
	4	El sistema elimina la huella y todos sus datos asociados de la base de datos.
Excepciones	Paso	Acción
	1, 3	El actor usuario cancela la operación. El caso de uso queda sin efecto

CU-016	Ver detalles de un individuo	
Versión	2.0 (15/06/2016)	
Autores	Mario Rodríguez García	
Descripción	El sistema deberá permitir al actor usuario ver los detalles de un individuo de la base de datos del sistema según se describe en el siguiente caso de uso.	
Precondición	El usuario debe estar en la pantalla de gestión de individuos.	
Secuencia normal	Paso	Acción
	1	El actor usuario selecciona el botón <i>Ver</i> en la pantalla de gestión de individuos.
	2	El sistema muestra en una nueva ventana los datos del individuo.
Excepciones	Paso	Acción
	1	El actor usuario cancela la operación. El caso de uso queda sin efecto.

CU-017	Ver imagen original de una huella dactilar	
Versión	2.0 (15/06/2016)	
Autores	Mario Rodríguez García	
Descripción	El sistema deberá permitir al actor usuario ver la imagen original de una huella dactilar según se describe en el siguiente caso de uso.	
Precondición	El usuario debe estar en la pantalla de creación de verificación de huellas dactilares, identificación de un individuo o en los detalles de un individuo.	
Secuencia normal	Paso	Acción
	1	El actor usuario selecciona el botón <i>Imagen Original</i> en la pantalla de detalles de individuo, en la de verificación de dos huellas dactilares o en la de identificación de un individuo.
	2	El sistema muestra por pantalla la imagen original de la huella actualmente seleccionada.
Excepciones	Paso	Acción
	1	El actor usuario cancela la operación. El caso de uso queda sin efecto.

CU-018	Ver imagen con minutiaes resaltados de una huella dactilar	
Versión	2.0 (15/06/2016)	
Autores	Mario Rodríguez García	
Descripción	El sistema deberá permitir al actor usuario ver la imagen con los minutiaes resaltados según se describe en el siguiente caso de uso.	
Precondición	El usuario debe estar en la pantalla de creación de verificación de huellas dactilares, identificación de un individuo o en los detalles de un individuo.	
Secuencia normal	Paso	Acción
	1	El actor usuario selecciona el botón <i>Imagen con Minutiaes</i> en la pantalla de detalles de individuo, en la de verificación de dos huellas dactilares o en la de identificación de un individuo.
	2	El sistema muestra por pantalla la imagen con los minutiaes resaltados de la huella actualmente seleccionada.
Excepciones	Paso	Acción
	1	El actor usuario cancela la operación. El caso de uso queda sin efecto.

CU-019	Ver imagen binaria de una huella dactilar	
Versión	2.0 (15/06/2016)	
Autores	Mario Rodríguez García	
Descripción	El sistema deberá permitir al actor usuario ver la imagen binaria de una huella dactilar según se describe en el siguiente caso de uso.	
Precondición	El usuario debe estar en la pantalla de creación de verificación de huellas dactilares, identificación de un individuo o en los detalles de un individuo.	
Secuencia normal	Paso	Acción
	1	El actor usuario selecciona el botón <i>Imagen Binarizada</i> en la pantalla de detalles de individuo, en la de verificación de dos huellas dactilares o en la de identificación de un individuo.
	2	El sistema muestra por pantalla la imagen binarizada de la huella actualmente seleccionada.
Excepciones	Paso	Acción
	1	El actor usuario cancela la operación. El caso de uso queda sin efecto.

## 2.2 Diagrama de clases

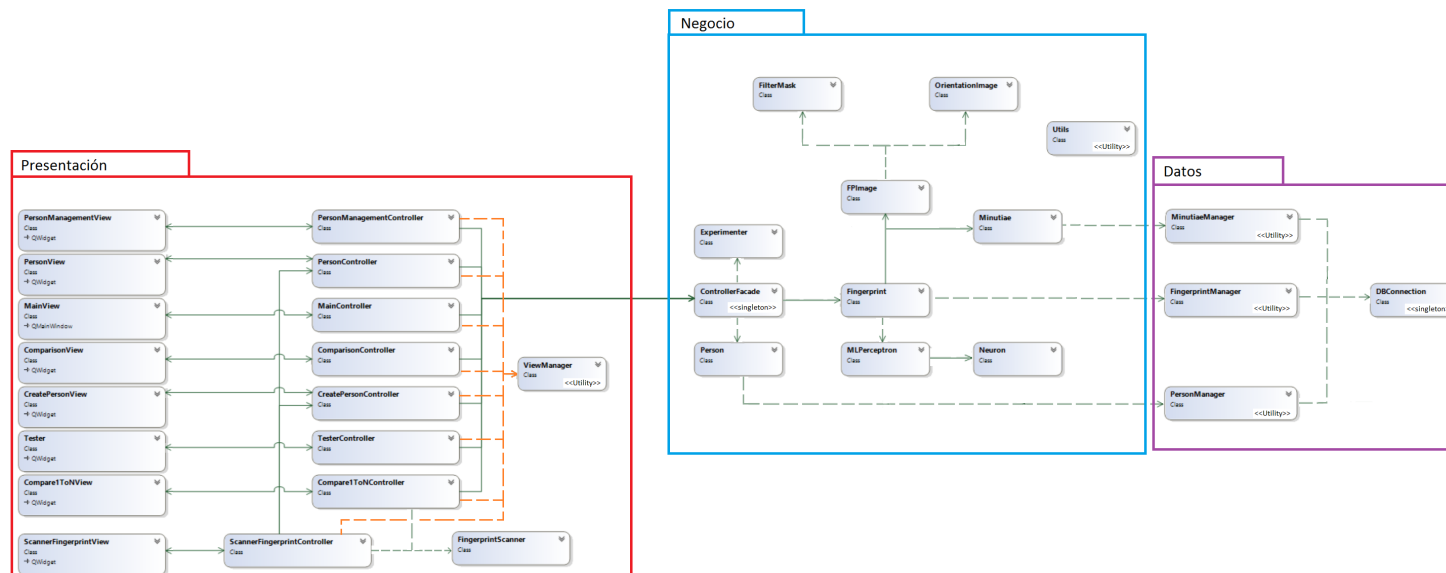
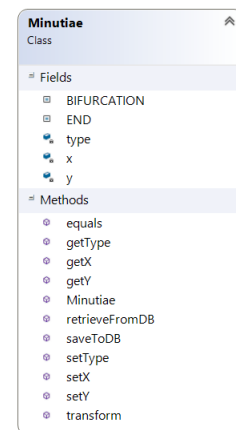
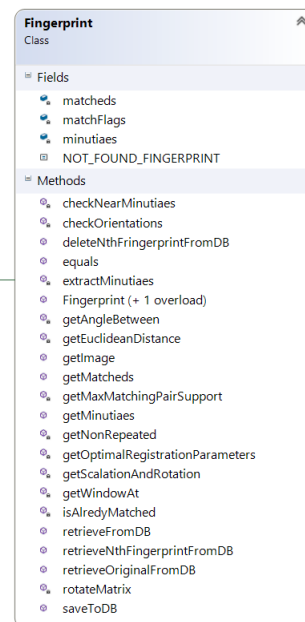
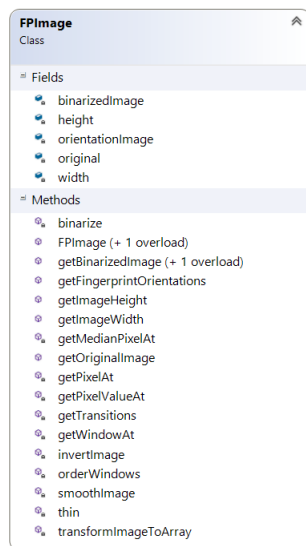
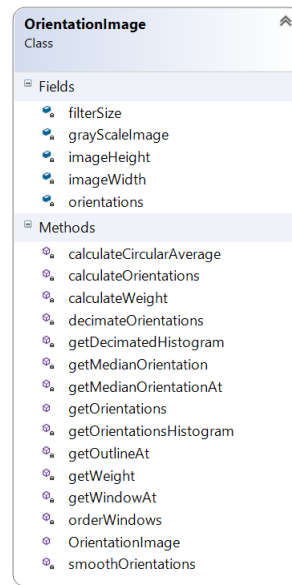
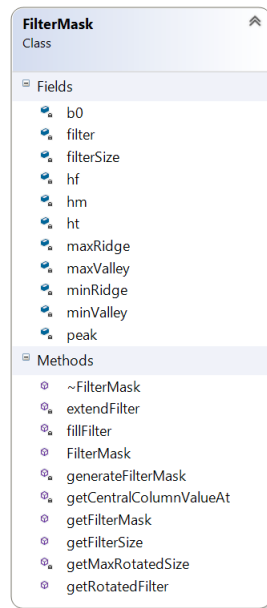
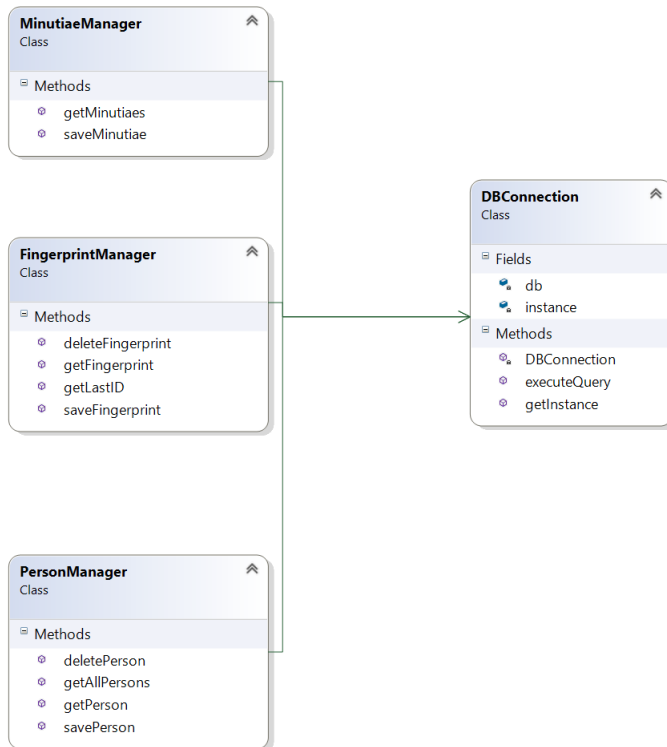
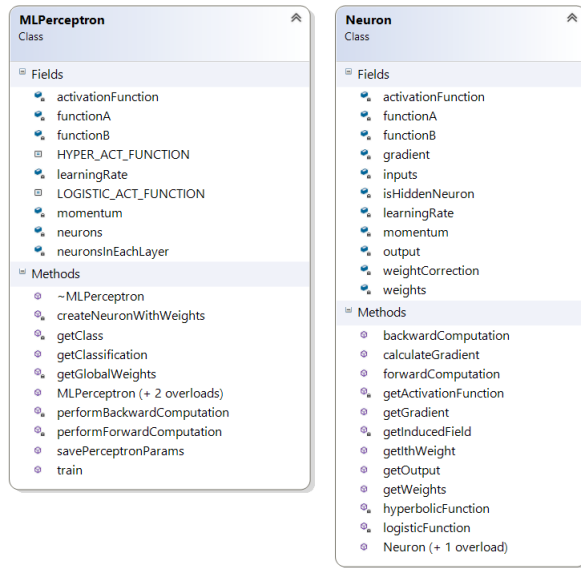
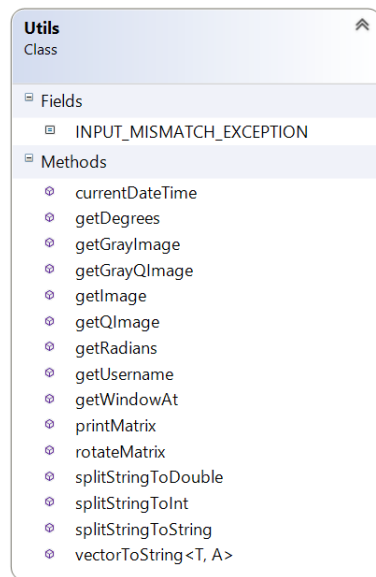
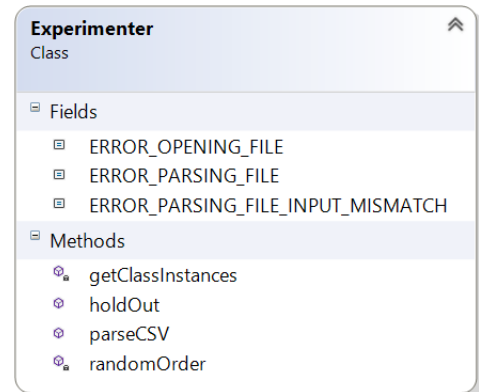
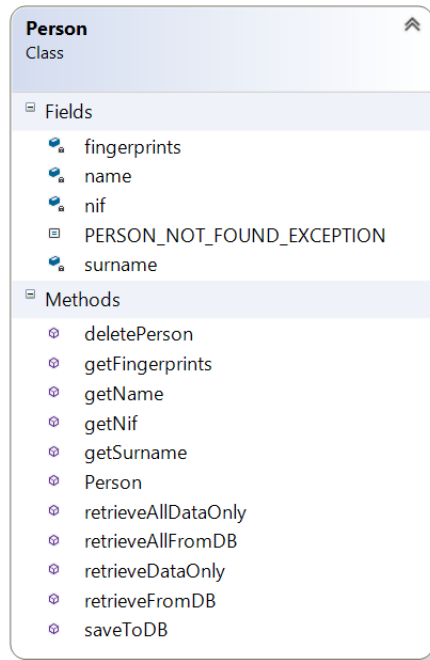


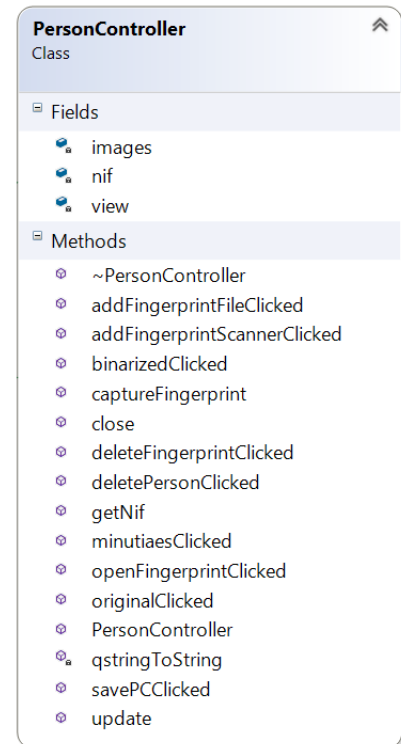
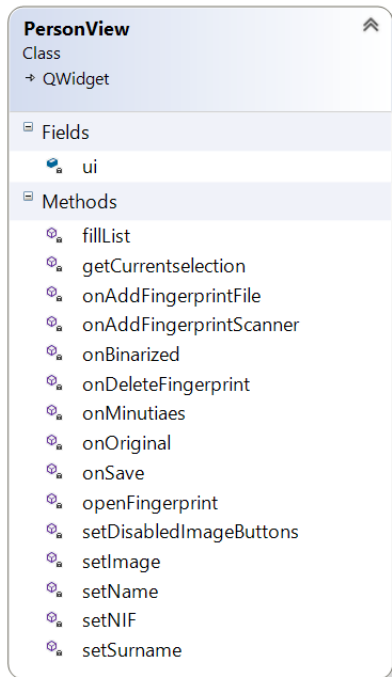
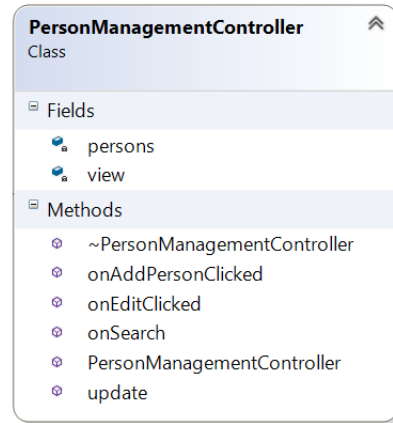
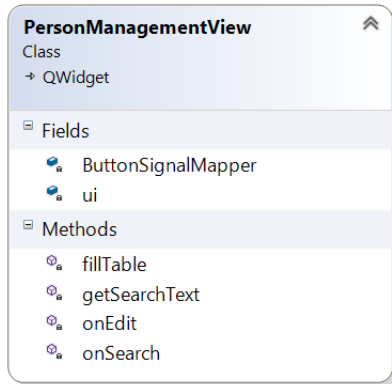
Figura 2.2: Diagrama de clases de diseño del sistema desarrollado.

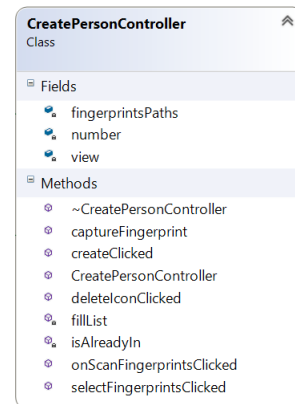
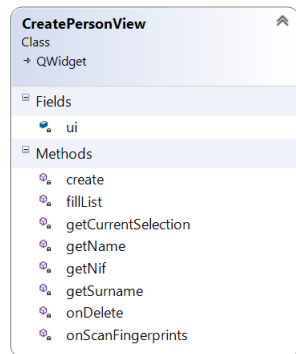
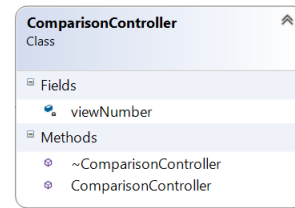
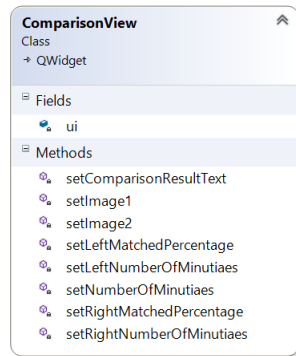
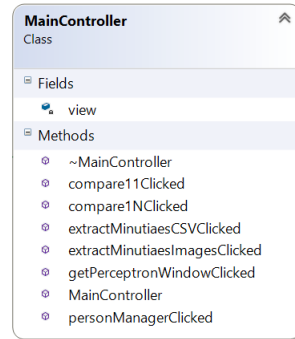
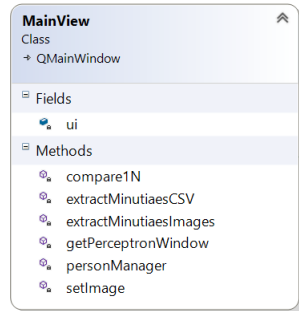












**Tester**  
Class  
↳ QWidget

Fields

- ui

Methods

- getEfficiency
- getEpochs
- getFunctionA
- getFunctionB
- getLearningRate
- getMomentum
- getNeurons
- getPath
- getTrainingPercent
- setPathText

**TesterController**  
Class

Fields

- view

Methods

- ~TesterController
- getEfficiencyClicked
- getInstancesClicked
- TesterController

**Compare1ToNView**  
Class  
↳ QWidget

Fields

- ui

Methods

- changeUi
- checkObtainL
- checkObtainR
- clearImage2
- disableCompareButton
- disableLeftButtons
- disableRightButtons
- getImage1
- getImage2
- hideRightButtons
- hideRightCaptureButtons
- onBinarizeL
- onBinarizeR
- onCaptureL
- onCaptureR
- onCompare
- onExtractMinutiaeL
- onExtractMinutiaeR
- onLoadR
- onOriginalL
- onOriginalR
- onSavePCL
- onSavePCR
- setCheckObtainL
- setCheckObtainR
- setImage1
- setImage2
- setTextLoadButtonL
- setTextLoadButtonR

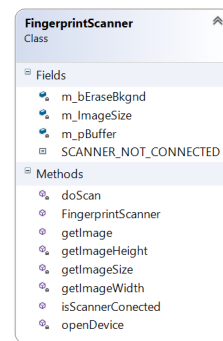
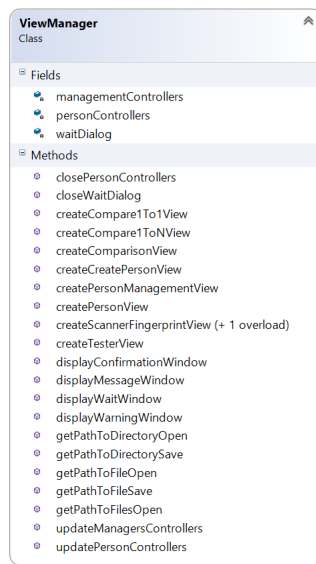
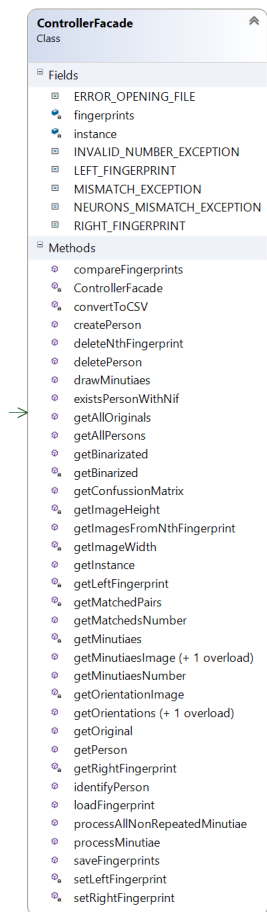
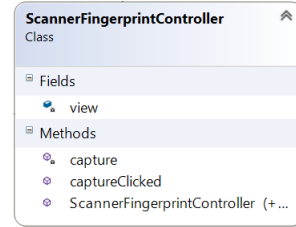
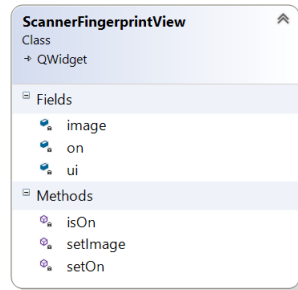
**Compare1ToNController**  
Class

Fields

- loadedF1
- loadedF2
- number
- oneToN
- view
- viewNumber

Methods

- ~Compare1ToNController
- captureL
- captureR
- Compare1ToNController
- displayScannerError
- onBinarizeLClicked
- onBinarizeRClicked
- onCaptureLClicked
- onCaptureRClicked
- onCompareClicked
- onExtractMinutiaeLClicked
- onExtractMinutiaeRClicked
- onLoadLClicked
- onLoadRClicked
- onOriginalLClicked
- onOriginalRClicked
- onSavePCLClicked
- onSavePCRClicked



### 2.3 Diagrama entidad relación

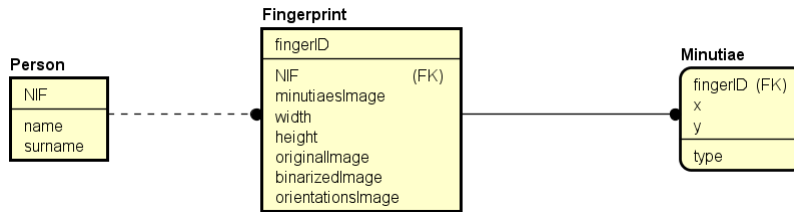


Figura 2.3: Diagrama entidad relación de la base de datos.

### 2.4 Diagrama relacional

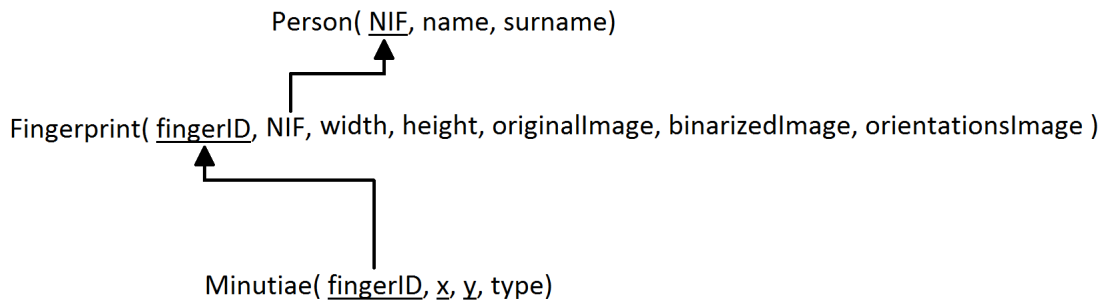


Figura 2.4: Diagrama relacional de la base de datos.





## Capítulo 3

# Binarización

La binarización de una huella dactilar consiste en la conversión de una imagen en escala de grises en una en blanco y negro exclusivamente, o binaria. Este proceso ayuda a eliminar el ruido existente en la imagen original, como las rupturas de líneas o zonas difusas debido a su mala calidad. El algoritmo de binarización descrito en éste capítulo está basado en las publicaciones científicas [6] y [7].

El primer paso del proceso es el diseño del filtro para el dominio espacial. En este trabajo se ha elegido un filtro gaussiano, que elimina el ruido y parte de la información innecesaria. Tras el cálculo del filtro, o máscara de filtro, lo siguiente que se debe realizar es el cálculo de la imagen de orientaciones, que almacena la orientación de la huella dactilar en cada píxel. Con la máscara de filtro y la imagen de orientaciones se realiza un procesamiento píxel a píxel para determinar su color: blanco o negro. Tras su ejecución, se obtiene una imagen completamente binaria; sin embargo, aún posee ruido que debe ser eliminado aplicando un filtro del dominio espacial mediano sobre la imagen binaria.

### 3.1 Máscara de filtro

La máscara de filtro depende de cinco parámetros que han de ser prefijados por el desarrollador. Estos parámetros son:

- El valor mínimo del ancho de una cresta:  $c_{\min}$
- El valor máximo del ancho de una cresta:  $c_{\max}$
- El valor mínimo del ancho de un valle:  $v_{\min}$
- El valor máximo del ancho de un valle:  $v_{\max}$
- El valor máximo del filtro:  $a_0$

A partir de estos parámetros se calcula el tamaño del filtro  $k$ .

El filtro diseñado tiene que cumplir tres condiciones: ser cuadrado, es decir, sus dimensiones son  $k \times k$ , ser simétrico e impar. El valor de  $k$  es el del periodo de la señal y debe ser impar. El periodo de la señal es el correspondiente a un valle y una cresta de la huella dactilar para la que se está diseñando el filtro.

El filtro está orientado horizontalmente y compuesto por tres tipos de franjas: central, laterales y las de transición. La primera contiene los coeficientes que amplifican el valor de las crestas. Las laterales están formadas por los coeficientes que amplifican el valor de los valles. Y las de transición contienen coeficientes nulos, que sirven de unión entre las franjas laterales y la central. El filtro  $f(i, j)$ , donde  $i$  es el índice de las filas;  $j$ , el de las columnas, y sus valores centrales son  $i = 0$  y  $j = 0$ , tiene las franjas definidas de la siguiente manera:

$$\begin{aligned} \text{Franja central:} & \quad -h_m \leq j \leq h_m \\ \text{Franjas de transición:} & \quad -h_t < j < -h_m; h_m < j < h_t \\ \text{Franjas laterales:} & \quad -h_f \leq j \leq -h_t; h_t \leq j \leq h_f \end{aligned} \quad (3.1)$$

Donde  $h_m$ ,  $h_t$  y  $h_f$  son las distancias desde el centro de cada franja a sus respectivo final. Las ecuaciones para calcularlas son:

$$\begin{aligned} h_m &= \left[ \frac{c_{\min} - 1}{2} \right]_- \\ h_t &= \left[ \frac{c_{\max} + 1}{2} \right]_+ \\ h_m &= h_t - 1 + \left[ \frac{v_{\min} + 1}{2} \right]_+ \end{aligned} \quad (3.2)$$

El subíndice negativo en el corchete indica el truncamiento del valor real a un valor entero, y el subíndice positivo indica el redondeo a la alza de un valor real a uno entero. Una vez obtenidos estos parámetros podemos calcular el tamaño del filtro,  $k$ , mediante la ecuación siguiente:

$$k = 2h_f + 1 \quad (3.3)$$

Los coeficientes del filtro son generados a partir de los valores de la columna central,  $f(0, j)$ . Al coeficiente central de dicha columna,  $f(0, 0)$  se le asigna el valor  $a_0$ . El resto de la columna se obtiene atenuando este valor mediante la función coseno hasta conseguir un mínimo de  $a_0/\sqrt{2}$  en el punto  $j = \pm h_m$ . Como ya mencionamos anteriormente, la franja de transición se llena con ceros. El máximo de la franja lateral se alcanza en  $j = \pm h_f$ , y se atenúa igualmente mediante la función coseno hasta obtener un valor igual a  $b_0/\sqrt{2}$  en  $j = \pm h_t$ .

Teniendo en cuenta estas condiciones, la columna central del filtro,  $f(0, j)$  es calculada mediante la siguiente función a trozos:

$$f(0, j) = f(0, -j) = \begin{cases} a_0 \cos \frac{j\pi}{4h_m} & 0 < j \leq h_m \\ 0 & h_m < j < h_t \\ b_0 \cos \frac{\pi(h_f - j)}{4(h_f - h_t)} & h_t \leq j \leq h_f \end{cases} \quad (3.4)$$

Donde el valor de  $b_0$  es hallado igualando la suma de los coeficientes de la columna central a 0:

$$a_0 + 2 \sum_{j=1}^{h_m} f(0, j) + 2 \sum_{j=h_t}^{h_f} f(0, j) = 0 \quad (3.5)$$

Y sustituyendo  $f(0, j)$  en la ecuación 3.4 obtenemos:

$$b_0 = \frac{-a_0 \left[ 1 + 2 \sum_{j=1}^{h_m} \cos \frac{j\pi}{4h_m} \right]}{2 \sum_{j=h_t}^{h_f} \cos \left[ \frac{\pi(h_f - j)}{4(h_f - h_t)} \right]} \quad (3.6)$$

Tras calcular los coeficientes de la columna central con las ecuaciones 3.4 y 3.6, el resto del filtro se obtiene en función de la columna central con ayuda de la fórmula siguiente:

$$f(i, j) = f(-i, j) = f(i, -j) = f(-i, -j) = f(0, j) \cos \frac{i\pi}{3h_f} \quad (3.7)$$

Los parámetros empleados en este trabajo para el diseño del filtro son:

$$v_{\min} = 2 \quad v_{\max} = 4 \quad c_{\min} = 3 \quad c_{\max} = 5 \quad a_0 = 1000 \quad (3.8)$$

Esta elección se ha basado en que dichos valores son los que mejores resultados han proporcionado experimentalmente. El filtro que se genera con estos parámetros es el siguiente:

-250	-321	-383	-433	-469	-492	-500	-492	-469	-433	-383	-321	-250
-353	-454	-541	-612	-664	-696	-707	-696	-664	-612	-541	-454	-353
-353	-454	-541	-612	-664	-696	-707	-696	-664	-612	-541	-454	-353
-250	-321	-383	-433	-469	-492	-500	-492	-469	-433	-383	-321	-250
0	0	0	0	0	0	0	0	0	0	0	0	0
353	454	541	612	664	696	707	696	664	612	541	454	353
500	642	766	866	939	984	1000	984	939	866	766	642	500
353	454	541	612	664	696	707	696	664	612	541	454	353
0	0	0	0	0	0	0	0	0	0	0	0	0
-250	-321	-383	-433	-469	-492	-500	-492	-469	-433	-383	-321	-250
-353	-454	-541	-612	-664	-696	-707	-696	-664	-612	-541	-454	-353
-353	-454	-541	-612	-664	-696	-707	-696	-664	-612	-541	-454	-353
-250	-321	-383	-433	-469	-492	-500	-492	-469	-433	-383	-321	-250

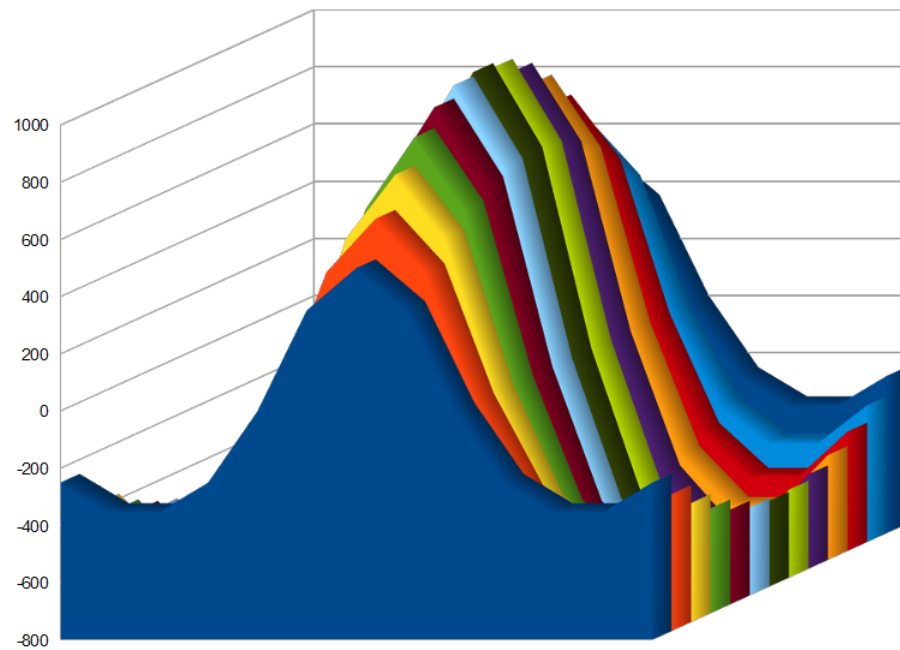


Figura 3.1: Imagen en 3D de los valores del filtro generado.

En la imagen tridimensional de la figura 3.1, se puede apreciar lo bien que se ajusta el filtro a una cresta de una huella dactilar. Sin embargo, sólo se podría aplicar en crestas orientadas horizontalmente, y para que el filtro pueda ser usado en todas, independientemente de su orientación, deberían ser rotadas un ángulo  $\theta$ , igual al de la cresta en ese punto. El coeficiente localizado en

$(i', j')$  del filtro rotado se obtiene girando un ángulo  $\theta$  hacia la posición  $(i, j)$  en el filtro original:

$$\begin{bmatrix} i \\ j \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} i' \\ j' \end{bmatrix} \quad (3.9)$$

Normalmente  $(i, j)$  no va a ser exactamente una posición del filtro original. Cuando esto sucede, el coeficiente es calculado como una interpolación de Lagrange de los cuatro coeficientes más cercanos a  $(i, j)$ , que se denotan como:  $(i_L, j_L)$ ,  $(i_U, j_U)$ ,  $(i_L, j_U)$ ,  $(i_U, j_L)$ , donde los subíndices L y U indican la posición de abajo y la de arriba respectivamente de los cuatro coeficientes más cercanos a  $(i, j)$ , por lo tanto:

$$\begin{aligned} f(i', j') = & (j_U - j')(i_U - i')f(i_L, j_L) + (i' - i_L)(j_U - j')f(i_U, j_L) \\ & + (j' - j_L)(i_U - i')f(i_L, j_U) + (j' - j_L)(i' - i_L)f(i_U, j_U) \end{aligned} \quad (3.10)$$

Las ecuaciones 3.9 y 3.10 permiten la rotación del filtro para poder aplicarlo en todas las orientaciones posibles. Para un filtro de tamaño  $k \times k$  y orientación  $\theta$  usaremos la notación :  $f(k, \theta, (i, j))$ .

## 3.2 Imagen de orientaciones

La imagen de orientaciones es la matriz de las mismas dimensiones que la imagen de la huella dactilar. En ella se almacenan las orientaciones de las crestas en cada píxel  $(x, y)$ , que se obtiene calculando los pesos de orientación,  $p(\theta)$ , perpendiculares a tres ángulos:  $\theta \in \{0, \pi/3, 2\pi/3\}$  y su media circular, también denominada orientación dominante.

Para la orientación horizontal ( $\theta = 0$ ) calculamos el peso como:

$$p(\theta = 0) = \sum_{x=0}^{k-1} \left[ \sum_{y=1}^{k-1} |s(x, y) - s(x, y-1)| \right] \quad (3.11)$$

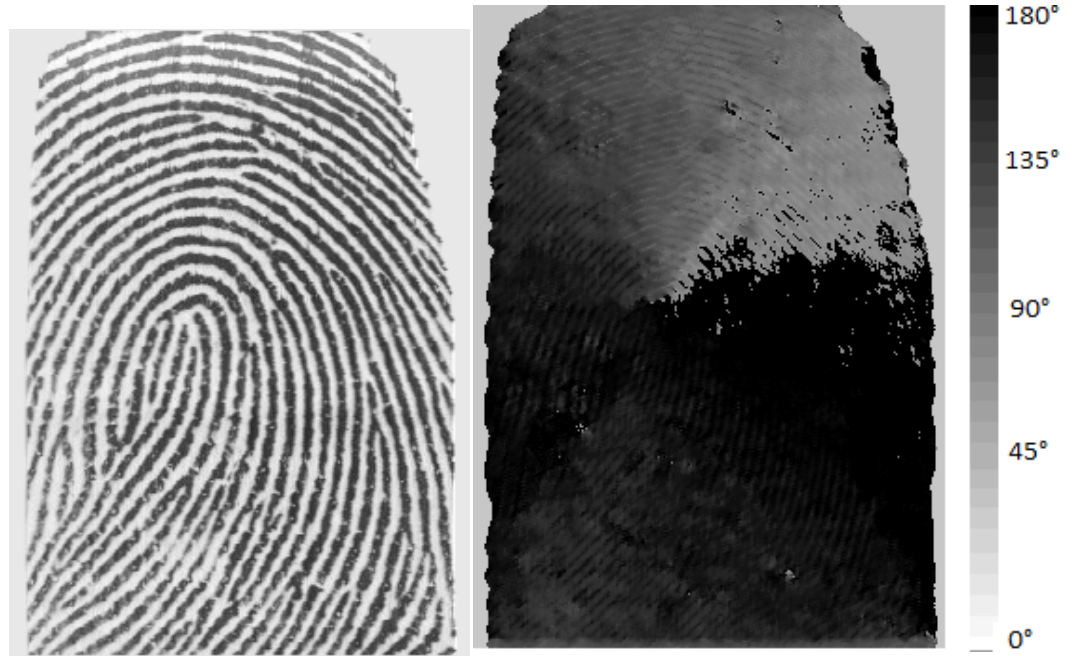
Donde  $s(x, y)$  es una matriz cuadrada centrada en  $(x, y)$  con dimensiones  $k \times k$ . El cálculo de los pesos para las dos orientaciones restantes:  $p(\pi/3)$  y  $p(2\pi/3)$  se realiza haciendo uso de la ecuación 3.11 también, pero se han de rotar las matrices:  $s(x, y)$  y  $s(x, y-1)$ ,  $\pi/3$  y  $2\pi/3$  respectivamente. Para evitar que al rotar las matrices  $s(x, y)$  y  $s(x, y-1)$  haya posiciones fuera del rango  $k \times k$ , las dimensiones iniciales de las matrices  $s(x, y)$  y  $s(x, y-1)$  deben ser de  $2k \times 2k$ . Tras realizar la rotación, se recortan los extremos de la matriz para obtener las dimensiones originales de  $k \times k$ .

Finalmente, la orientación en cada píxel es calculada realizando la media circular de los pesos:  $p(0)$ ,  $p(\pi/3)$  y  $p(2\pi/3)$ .

$$\theta_{\text{MED}} = \begin{cases} \theta_{\text{max}} + \frac{1}{2} \frac{p(\theta_{\text{mid}} - p(\theta_{\text{min}}))}{p(\theta_{\text{max}} - p(\theta_{\text{min}}))} & \text{si } (\theta_{\text{max}} + 1)\pi = \theta_{\text{mid}} \\ \theta_{\text{max}} - \frac{1}{2} \frac{p(\theta_{\text{mid}} - p(\theta_{\text{min}}))}{p(\theta_{\text{max}} - p(\theta_{\text{min}}))} & \text{si } (\theta_{\text{max}} - 1)\pi = \theta_{\text{mid}} \end{cases} \quad (3.12)$$

Donde  $p(\theta_{\text{max}}) \geq p(\theta_{\text{mid}}) \geq p(\theta_{\text{min}})$ . El subíndice  $(\cdot)_{\pi}$  indica la operación módulo  $\pi$ . Si no se encuentra ninguna orientación dominante, el valor de la orientación en ese píxel es  $-1$  a modo de bandera.

Una vez aplicada esta técnica en todos los píxeles, obtendremos como resultado una imagen como la de la figura 3.2b, donde el nivel de intensidad del gris en un punto indica el ángulo de la cresta en esa posición.



(a) Huella dactilar original.

(b) Imagen de orientaciones de la huella dactilar de la izquierda.

Figura 3.2: Ejemplo del proceso de extracción de la imagen de orientaciones de una huella dactilar.

La imagen de orientaciones resultante posee bastantes irregularidades, dado que en zonas pequeñas se producen grandes cambios de orientaciones, y en algunas zonas muy pequeñas no se ha detectado ninguna. Para corregir estas

irregularidades, se reduce el número de orientaciones totales y se realiza un suavizado de la imagen aplicando un filtro mediano con un tamaño  $7 \times 7$ .

Para disminuir el número de orientaciones finales, lo primero es elegir el número final de ellas; en el caso de este trabajo, a 18. El siguiente paso es el cálculo del histograma, consistente en un vector con el número de veces que se repite cada orientación. A partir del histograma, se elige la más repetida cada 10 grados, es decir, la más usada de los 0 a los 10 grados, de los 10 a los 20 grados, etc. Tras obtener las 18 más usadas, se sustituyen las orientaciones originales por la más cercana a una de las 18 seleccionadas.

El siguiente paso es aplicar un filtro mediano de tamaño de ventana  $7 \times 7$  a la imagen de orientaciones. Este proceso elimina la mayor parte del ruido y suaviza los cambios de orientaciones, ayudando en la fase posterior del proceso de binarización a obtener mejores resultados.

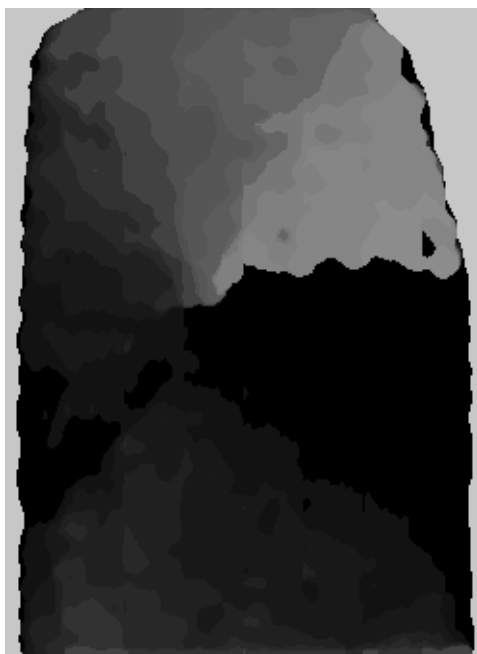


Figura 3.3: Imagen del resultado del suavizado en la imagen de orientaciones de la huella de la figura 3.2a.

### 3.3 Procesamiento píxel a píxel

El último paso del proceso de binarización es el procesamiento píxel a píxel, en el cual se aplica el filtro con la rotación indicada en la imagen de orientaciones,

calculada en el paso anterior, en todos los píxeles de la imagen. El píxel  $(x, y)$  de la imagen binarizada se calcula mediante la función siguiente:

$$(x, y) = \begin{cases} 0 & \text{si } s'(x, y) \leq 0 \\ 1 & \text{si } s'(x, y) > 0 \end{cases} \quad (3.13)$$

Donde  $s'(x, y)$  se consigue aplicando la ecuación:

$$s'(x, y) = \sum_{j=\frac{k-1}{2}}^{\frac{k-1}{2}} \sum_{i=\frac{k-1}{2}}^{\frac{k-1}{2}} s(x+i, y+j) f(k, \theta, (i, j)) \quad (3.14)$$

Siendo  $\theta$  el ángulo contenido en la imagen de orientaciones en el píxel  $(x, y)$ . Si el valor de  $\theta$  en algún píxel  $(x, y)$  es  $-1$ , ese píxel pasa a valer automáticamente 1 sin necesidad de hacer cálculo alguno, dado que el valor  $-1$  en la imagen de orientaciones indica que no hay orientación en ese píxel, y por lo tanto no existe cresta en él.



(a) Huella dactilar original.

(b) Imagen binarizada.

Figura 3.4: Comparación entre la imagen original y la binarizada.



En la imagen de la figura 3.4b, se puede apreciar que el resultado del proceso de binarización de la huella dactilar genera unos resultados muy buenos; sin embargo, aún es posible optimizar este resultado aplicando un filtro mediano con dimensiones  $3 \times 3$ . Este filtro mediano suavizará las líneas y unirá muchas de ellas, que, debido a la posible mala calidad de la imagen de la huella dactilar, estaban separadas. Los resultados de aplicar el filtro se pueden ver en la figura 3.5.



Figura 3.5: Resultado del suavizado sobre la imagen binarizada de la figura 3.4b.



## Capítulo 4

# Adelgazamiento

El proceso de adelgazamiento de una imagen binarizada de una huella dactilar consiste en limitar la anchura de las líneas a un tamaño fijo de un píxel de grosor. Esto sirve para eliminar gran cantidad de información innecesaria, ya que la mayoría de píxeles negros no aportan conocimiento a la red neuronal. El algoritmo de adelgazamiento descrito en este capítulo está basado en el paper [5].

Para el adelgazamiento, se evalúa cada píxel negro de la imagen, es decir, los pertenecientes a las crestas de la huella y sus píxeles vecinos. Si el píxel evaluado cumple una serie de condiciones, su color pasa de negro a blanco, o lo que es lo mismo, es borrado de la imagen. La vecindad de un píxel se establece según muestra la siguiente tabla:

P9	P2	P3
P8	P1	P4
P7	P6	P5

Donde P1 es el píxel a evaluar y el resto son sus vecinos, con los que puede estar conectado de dos maneras: tetra-conectado, cuando todos sus vecinos horizontales y verticales tiene valor 1:

	1	
1	P1	1
	1	

Table 4.1: Ejemplo de tetra-conexión.

Y octa-conectado, cuando todos sus vecinos son unos:

1	1	1
1	P1	1
1	1	1

Table 4.2: Ejemplo de octa-conexión.

De entre los algoritmos posibles para el adelgazamiento de imágenes binarias, hemos escogido el de Guo-Hall, ya que es el que proporciona el resultado de más calidad con respecto a otros algoritmos a cambio de una ínfima pérdida de eficiencia.

## 4.1 Algoritmo Guo-Hall

El algoritmo de Guo-Hall es un algoritmo iterativo, en el que, en cada pasada, se borran parte de los píxeles de todas las líneas. Se ejecuta hasta que no sea posible eliminar ninguno más, esto es, la imagen permanezca inalterada tras la ejecución de una iteración.

El pseudocódigo del algoritmo de Guo-Hall es el siguiente:

```

invertir imagen binarizada

while haya puntos que borrar do
  for all pixeles p(i,j) do
    if (a) C(P1) = 1;
       (b) 2 <= N(P1) <= 3;
       (c) Aplicar una de las siguientes:
           1. (P2 | P3 | !P5) | P4 = 0 en iteraciones impares
           2. (P6 | P7 | !P9) & P8 = 0 en iteraciones pares
    then
      Borrar pixel p(i,j)
    end if
  end for
end while

invertir imagen binarizada

```

Siendo  $C(P1)$  la condición que conserva la conectividad de las líneas y que evita la eliminación de píxeles que provocarían una interrupción de la línea. Esta condición se define como el número de vecinos que son unos y están octa-conectados, y es calculada como:

$$C(P1) = \bar{P}2 \wedge (P3 \vee P4) + \bar{P}4 \wedge (P5 \vee P6) + \bar{P}6 \wedge (P7 \vee P8) + \bar{P}8 \wedge (P9 \vee P2) \quad (4.1)$$

La variable  $N(P1)$  se encarga de detectar los finales de línea y se obtiene con las fórmulas siguientes:

$$N(P1) = MIN\{N_1(P1), N_2(P1)\} \quad (4.2)$$

$$N_1(P1) = (P9 \vee P2) + (P3 \vee P4) + (P5 \vee P6) + (P7 \vee P8) \quad (4.3)$$

$$N_2(P1) = (P2 \vee P3) + (P4 \vee P5) + (P6 \vee P7) + (P8 \vee P9) \quad (4.4)$$

Finalmente, hay dos condiciones que se aplican en función de si la iteración es par o impar. En el caso de las pares, borra los vecinos de la parte noreste del píxel, y en las impares, elimina los vecinos de la parte suroeste del píxel.

El resultado de aplicar el algoritmo de Guo-Hall sobre la imagen binarizada obtenida en el capítulo 3 se puede apreciar en la figura siguiente.



Figura 4.1: Resultado del adelgazamiento sobre la imagen binarizada de la figura 3.5.



## Capítulo 5

# Extracción de minutiae

Tras reducir todo lo posible el ruido y la información innecesaria con los pasos de binarización y adelgazamiento, se procederá a la extracción de los minutiae de la huella dactilar adelgazada. Aunque existen doce tipos de minutiae, en este trabajo sólo se consideran dos de ellos: las bifurcaciones y los fines de línea, porque el resto de tipos son casos especiales de éstos. La información necesaria de un minutiae para el proceso de reconocimiento de una huella dactilar son sus coordenadas  $(x, y)$  en la huella y su tipo. Con el fin de obtener dicha información, necesitamos ser capaces de reconocer el minutiae en la huella y su tipo automáticamente. Este es un problema que se denomina reconocimiento de patrones. En este trabajo ha sido resuelto con increíble precisión y eficacia por las Redes Neuronales Artificiales. Las aportaciones descritas en este capítulo están basadas en el libro [3].

### 5.1 Red Neuronal: Perceptrón Multicapa

En este trabajo, el tipo de red neuronal escogido es el Perceptrón Multicapa, debido a su buen funcionamiento y popularidad para resolver problemas de reconocimiento de patrones. Como puede verse en la figura 5.1, su estructura consta de una capa de entrada, un conjunto de capas ocultas y una capa de salida, por lo tanto, se organiza en capas. De este modo, la conexión entre neuronas de la misma capa es nula y sólo existe interacción entre los nodos de una determinada capa con los de la anterior y posterior, si fuese el caso.

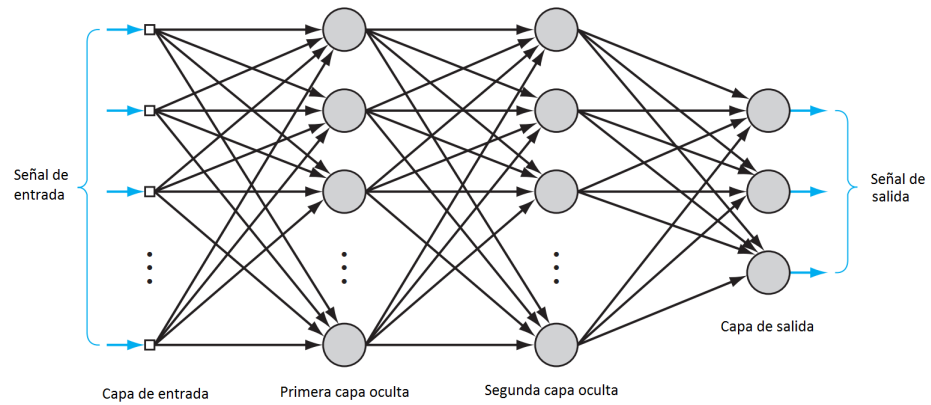


Figura 5.1: Arquitectura de un perceptrón multicapa con dos capas ocultas.

Se han realizado varios experimentos con diferentes diseños de redes neuronales, con el fin de obtener el mejor rendimiento de la red neuronal en el problema de extracción de minutiae. El primer prototipo consiste en un solo perceptrón multicapa con tres neuronas en la capa de salida.

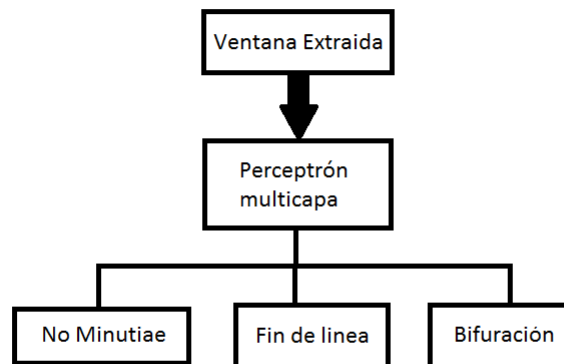


Figura 5.2: Primer diseño de la red neuronal para la extracción de minutiae.



El diseño propuesto es probado con un experimento de Hold-Out repetido diez veces al 66.7 %, y la tasa de acierto obtenida es de un 80 %. Sin embargo, esta tasa varía drásticamente entre las diferentes clases. La mayoría de los no minutiae son clasificados correctamente, pero tanto las bifurcaciones como los finales de línea tienen una tasa de acierto inferior al 10 %. Este hecho se debe a que las clases no están en la misma proporción, como en el caso de los no minutiae, cuya frecuencia de aparición es diez veces mayor que las bifurcaciones o los fines de línea. Por este motivo, se descarta este diseño y se crea una red neuronal especializada. Para ello, se dispone de un perceptrón multicapa para evaluar si la muestra es, o no, un minutiae. En el caso de que lo sea, la muestra será evaluada por otro perceptrón multicapa, que se encargará de discernir el tipo de minutiae del que se trata.

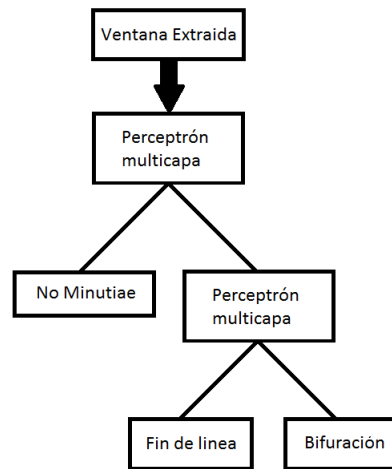


Figura 5.3: Red neuronal con especialización.

De nuevo, se pone a prueba este diseño de red neuronal con un experimento Hold-Out repetido diez veces al 66.7 %; la tasa de acierto obtenida es de un 33% en el caso del primer perceptrón multicapa, que distingue entre un minutiae y no minutiae, y de un 96.5 % en el caso del segundo perceptrón multicapa, que clasifica al minutiae en función de su tipo. Aunque el número de minutiae correctamente clasificados aumente notoriamente, a un 32.5 % en total, sigue siendo una porcentaje muy bajo para un AFIS. Esta baja tasa de aciertos se debe a que las clases siguen desproporcionadas, pese a dividir el experimento inicial en dos clases: minutiae y no minutiae. Para solucionar este problema se diseña de nuevo la red neuronal; en esta ocasión, sustituimos el primer perceptrón del experimento anterior por tantos perceptrones como desproporcionadas estén las clases; en este caso, los no minutiae tiene una proporción de 10:1 respecto a los minutiae, por lo tanto, se reemplaza por diez perceptrones multicapa y un sistema de votación que decidirá si el patrón es

clasificado como un minutiae o no. Estos nuevos perceptrones no son entrenados con todas las instancias de ambas clases, sino que se seleccionan aleatoriamente y sin solapamientos diez conjuntos de instancias de la clase no minutiae, los cuales están, aproximadamente, en la misma proporción que los de la clase minutiae, y cada uno de ellos es entrenado con todas las instancias de minutiae y solo con uno de los conjuntos creados anteriormente, diferente al escogido para el resto de los perceptrones.

El sistema de votación de esta nueva red neuronal clasifica la instancia como no minutiae, si dos o más de los perceptrones multicapa clasifican la instancia como tal. De lo contrario, se clasifica como minutiae, y su tipo es determinado por otro perceptrón multicapa, especializado en la detección del tipo de minutiae. El diseño de esta red neuronal se puede ver en la figura 5.4.

Tras someter esta red neuronal a un experimento Hold-Out, como las anteriores veces, se obtiene una tasa de acierto del 93 % a la hora de clasificar correctamente un patrón, lo que es un porcentaje bastante bueno para un sistema de verificación de huellas dactilares y, por lo tanto, no es necesaria ninguna mejora adicional.

Los parámetros usados para los perceptrones multicapa y la función de activación son:

$$\alpha = 0.01 \quad \eta = 0.2 \quad a = 1.7159 \quad b = 0.666 \quad (5.1)$$

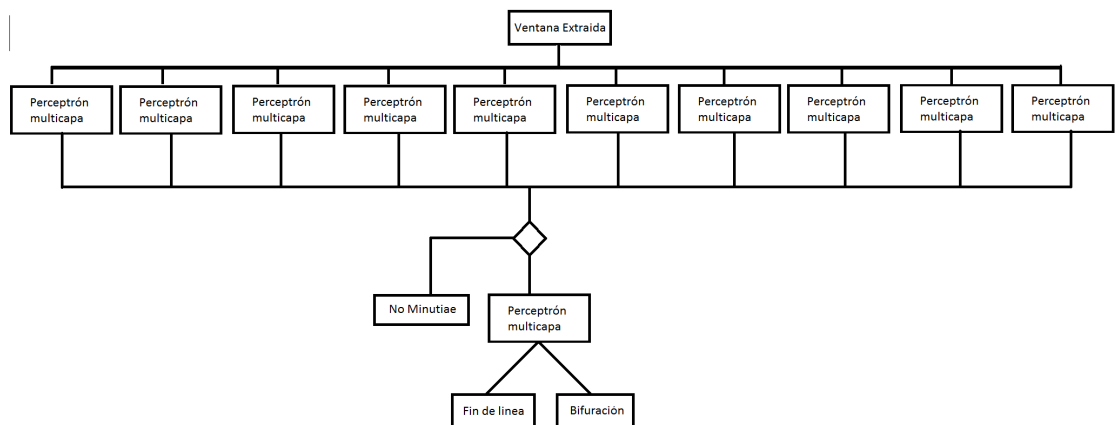


Figura 5.4: Red neuronal con especialización y sistema de votación.

## 5.2 Entrenamiento de la red neuronal

Los datos de entrenamiento de la red neuronal son divididos en tres clases diferentes: fines de línea, bifurcaciones y no minutias. Cada clase es representada en la red neuronal mediante una neurona en la capa de salida, por lo tanto, su tamaño será de tres.

Los datos de entrenamiento son ventanas de  $5 \times 5$  píxeles, obtenidas de imágenes binarizadas y adelgazadas de huellas dactilares. Algunos ejemplos de los diferentes tipos de patrones existentes en los datos pueden verse en la figura siguiente:

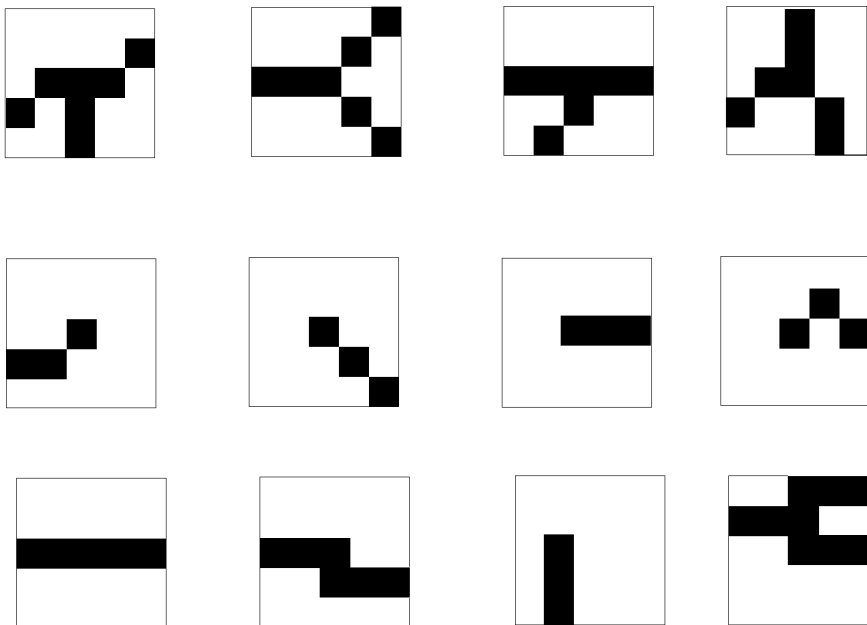


Figura 5.5: Diferentes ejemplos de patrones de bifurcaciones (primera fila), finales de línea (segunda fila) y no minutias (tercera fila).

Tal y como se aprecia en la figura anterior, los minutias se han seleccionado cuidadosamente para que coincidieran en el centro de la ventana; de lo contrario, estos serán considerados como no minutias.

Para entrenar al perceptrón multicapa se han procesado un total de 150 huellas dactilares. Tras ser procesadas, se han extraído de ellas las bifurcaciones, finales de línea y no minutias sin repetir. Todas estas ventanas, junto con sus respectivas clases, se han almacenado en un archivo csv. El resultado ha sido un total de 200 bifurcaciones, 51 finales de línea y 2778 no minutias.

### 5.3 Extracción de minutiae píxel a píxel

Los minutiae de una huella dactilar binarizada y adelgazada se obtienen evaluando la ventana centrada en cada píxel  $(x, y)$  de la imagen. Esta ventana se pasa como entrada a la red neuronal diseñada en la sección anterior. Para optimizar el rendimiento de la extracción de minutiae, las ventanas con el píxel central blanco serán clasificadas directamente como no minutiae, sin pasar por la red neuronal.

Una vez que todos los píxeles han sido examinados por la Red Neuronal Artificial, dispondremos de un vector con las coordenadas de todos los minutiae existentes en la huella dactilar, así como su tipo. El resultado de este proceso se puede ver en la figura 5.6.

En la imagen, se puede apreciar el buen rendimiento de la Red Neuronal Artificial para el reconocimiento de minutiae, dado que todos son encontrados y clasificados correctamente.

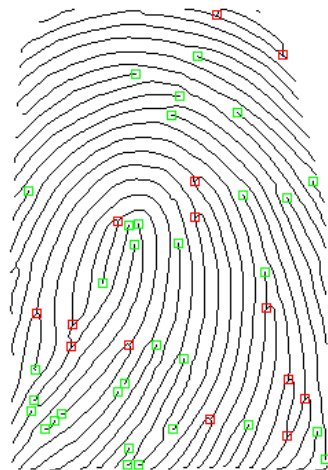


Figura 5.6: Ejemplo de huella dactilar binarizada y adelgazada con los minutiae resaltados.

## Capítulo 6

# Comparación de dos huellas dactilares

Los minutias de dos huellas dactilares pueden ser comparados mediante un algoritmo de patrones en dos dimensiones. Algunos de ellos cambian su tipo de una muestra a otra debido a la presión aplicada al tomar la huella dactilar o debido al proceso de binarización. Este cambio es especialmente común en el caso de las bifurcaciones, que, en muchas ocasiones tras el proceso de binarización, se transforman en fines de línea:



(a) Huella dactilar original.



(b) Imagen binarizada.

Figura 6.1: Ejemplo de la conversión de bifurcaciones en fines de línea en la fase de binarización.

Por este motivo, el tipo de minutiae queda descartado como atributo para la comparación de dos huellas dactilares y sólo se tiene en cuenta su posición absoluta  $(x, y)$ .

La comparación de dos huellas consiste en cotejar el conjunto de minutiae de la primera:  $P = \{p_1, p_2, \dots, p_m\}$ , con el de la segunda:  $Q = \{q_1, q_2, \dots, q_n\}$ , donde  $m$  y  $n$  son el número de minutiae detectados en cada huella respectivamente. El algoritmo de comparación de patrones bidimensionales empleado en este trabajo es invariante a la rotación o la escala de la huella dactilar. Estas propiedades resultan indispensables para tratar una misma huella obtenida de diferentes sensores, o usando el mismo aparato en instantes de tiempo diferentes. Por lo tanto, dos conjuntos de minutiae pueden tener una traslación, rotación o una escala diferente. El algoritmo de comparación tiene dos fases: la búsqueda del par principal,  $p_{\text{par.i}} \leftrightarrow q_{\text{par.a}}$  y la búsqueda de los pares coincidentes,  $p_i \leftrightarrow q_a$ . Los algoritmos explicados en éste capítulo están basados en la publicación científica [4].

## 6.1 Búsqueda del par principal

Consiste en encontrar un minutiae  $p_{\text{par.i}}$  en el conjunto  $P$ , que coincida con uno  $q_{\text{par.a}}$  del conjunto  $Q$ , y que bajo ese par,  $p_{\text{par.i}} \leftrightarrow q_{\text{par.a}}$ , exista el número máximo de pares  $p_i \leftrightarrow q_a$ . Esta cantidad se puede hallar a partir del par  $p_{\text{par.i}} \leftrightarrow q_{\text{par.a}}$  y se denota como  $c$ .

Para encontrar el par principal  $p_{\text{par.i}} \leftrightarrow q_{\text{par.a}}$ , se deben examinar las distancias euclídeas y los ángulos que existen entre todos los minutiae del mismo conjunto. A partir de las distancias euclídeas de los conjuntos  $P$  y  $Q$ , se obtiene la escala *a priori* respecto de la primera huella dactilar con la segunda. Los ángulos nos proporcionan la rotación existente entre ambas. Las fórmulas para el cálculo de las distancias, ángulos y la escala son las siguientes:

$$|\overrightarrow{p_i p_j}| = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \quad (6.1)$$

$$\theta_{\overrightarrow{p_i p_j}} = \begin{cases} \arctan\left(\frac{y_j - y_i}{x_j - x_i}\right) & \text{si } (x_j - x_i) > 0 \\ \arctan\left(\frac{y_j - y_i}{x_j - x_i}\right) + \pi & \text{si } (x_j - x_i) < 0 \\ \frac{\pi}{2} & \text{si } (x_j - x_i) = 0 \text{ y } (y_j - y_i) > 0 \\ -\frac{\pi}{2} & \text{si } (x_j - x_i) = 0 \text{ y } (y_j - y_i) < 0 \end{cases} \quad (6.2)$$

$$s = \frac{|\overrightarrow{q_a q_b}|}{|\overrightarrow{p_i p_j}|} \quad (6.3)$$

$$\theta = \theta_{\overrightarrow{q_a q_b}} - \theta_{\overrightarrow{p_i p_j}} \quad (6.4)$$

Donde tanto  $|\overrightarrow{q_a q_b}|$  como  $\theta_{\overrightarrow{q_a q_b}}$  se calculan de la misma forma que  $|\overrightarrow{p_i p_j}|$  y  $\theta_{\overrightarrow{p_i p_j}}$  respectivamente.

La escala  $s$  y el ángulo  $\theta$  son usados en la construcción del mapa  $M_{i,a}(s, \theta)$ , que almacena las ocurrencias de la tupla  $(s, \theta)$  para un par de minutias  $p_{\text{par.i}} \leftrightarrow q_{\text{par.a}}$ . El valor de ocurrencias de una tupla  $(s, \theta)$  indica el número de pares  $p_i \leftrightarrow q_a$  correctamente emparejados bajo esa escala y ese ángulo. El valor máximo del mapa  $M_{i,a}(s, \theta)$  indica el tope de pares  $p_i \leftrightarrow q_a$  que se van a emparejar correctamente si  $p_{\text{par.i}} \leftrightarrow q_{\text{par.a}}$  se adopta como el par principal. El algoritmo para calcular la escala, el ángulo y el número máximo de minutias coincidentes de un par  $p_{\text{par.i}} \leftrightarrow q_{\text{par.a}}$  es el que se muestra a continuación:

```

for j = 1, ..., m j ≠ i do
    calcular (|\overrightarrow{p_i p_j}|, \theta_{\overrightarrow{p_i p_j}})
end for

for b = 1, ..., n b ≠ a do
    calcular (|\overrightarrow{q_a q_b}|, \theta_{\overrightarrow{q_a q_b}})
end for

for j = 1, ..., m j ≠ i do
    for b = 1, ..., n b ≠ a do
        if banderas[j][b] = 1 then
            calcular (s, \theta)
            M_{i,a}(s, \theta)++
        end if
    end for
end for

(s_c, \theta_c) = \max(M_{i,a}(s, \theta)), y c = M_{i,a}(s_c, \theta_c)

```

La matriz banderas en el algoritmo indica si existe posibilidad de que el par  $p_j \leftrightarrow q_b$  sea coincidente y es inicializada con unos.

El algoritmo anteriormente descrito nos proporciona la escala y el ángulo únicamente, sin embargo, no nos da el par principal  $p_{\text{par.i}} \leftrightarrow q_{\text{par.a}}$ . Para ello, se emplea el siguiente algoritmo:

```

(pmax, s0, θ0, ppar-i, qpar-a) = (0, 0, 0, 0, 0)

for i = 1, ..., m do
  for a = 1, ..., n do
    calcular (c, sc, θc) del par pi ↔ qa

    if c < a then
      banderas[i][a] = 0
    end if

    if c < pmax then
      (pmax, s, θ, ppar-i, qpar-a) = (c, sc, θc, i, a)
    end if

    if pmax ≥ m - i then
      el par principal ha sido encontrado y el algoritmo finaliza
    end if
  end for
end for

```

El par  $p_{\text{par-}i} \leftrightarrow q_{\text{par-}a}$  se toma como el par principal y, la escala y la rotación, del conjunto de minutias  $P$  respecto del conjunto  $Q$ , son  $(s_0, \theta_0)$ .

## 6.2 Búsqueda de los pares coincidentes

La búsqueda de los pares coincidentes entre los conjuntos  $P$  y  $Q$  se realiza en dos pasos. El primero de ellos consiste en la búsqueda de los pares con escala y ángulo similares, es decir, el conjunto de pares cuya escala  $s$  cumpla la condición:  $|s - s_0| \leq \Delta s$  y cuyo ángulo verifique que  $|\theta - \theta_0| \leq \Delta \theta$ . Las cotas  $\Delta s$  y  $\Delta \theta$  son los límites aceptables para que la escala y el ángulo sean considerados similares respectivamente. Si los vectores  $|\overrightarrow{p_{\text{par-}i} p_j}|$  y  $|\overrightarrow{q_{\text{par-}a} q_b}|$  cumplen con las condiciones de similitud anteriormente descritas, entonces el par  $p_j \leftrightarrow q_b$  se mete en el conjunto de temporales. Si tras comparar el vector  $|\overrightarrow{p_{\text{par-}i} p_j}|$  con todos los del conjunto  $Q$ , el conjunto de temporales sólo posee un par, y es coincidente, entonces se mete en el de pares coincidentes. Sin embargo, hay ocasiones en las que dos o más pares cumplen esas condiciones debido a su cercanía. En ese caso, si el conjunto de temporales posee más de un único par, todos ellos se transfieren al conjunto de posibles.



El algoritmo para este primer paso es:

```

for j=1,...,m j≠i do
  set ib=0
  set temporales= []
  for b=1,...,n b≠a do
    if banderas[j][b] = 1 then
       $s = \frac{|\vec{q}_{par-a}q_b|}{|\vec{p}_{par-i}p_j|}$ 
       $\theta = \theta_{|\vec{q}_{par-a}q_b|} - \theta_{|\vec{p}_{par-i}p_j|}$ 
      if  $|s - s_0| \leq \Delta s$  &  $|\theta - \theta_0| \leq \Delta \theta$  then
        ib = b
        Agregar el par  $p_j \leftrightarrow q_b$  al conjunto de temporales
      end if
    end if
  end for
  if tamaño(temporales) = 1 then
    Agregar el par  $p_j \leftrightarrow q_b$  al conjunto de coincidentes
  elseif tamaño(temporales) > 1 then
    Agregar todos los pares del conjunto de temporales al conjunto de posibles
  end if
end for

```

Las medidas de similitud para la escala y ángulo,  $\Delta s$  y  $\Delta \theta$ , escogidas para este trabajo son:

$$\Delta s = 0.08 \qquad \Delta \theta = 3^\circ \qquad (6.5)$$

La elección de estos valores se ha realizado aplicando el método de ensayo y error, quedándose con los que mejores resultados proporcionaban. Valores más pequeños provocaban que bastantes de los minutiae que debían ser agregados al conjunto de coincidentes o posibles fueran desestimados, haciendo imposible avanzar en esta tarea en ciertas ocasiones.

El siguiente paso en el proceso de búsqueda de los pares coincidentes es la localización de aquellos que realmente coinciden en el conjunto de posibles. Para ello, se hace necesario transformarlos en:

$$p_{transf-j} = G_r(p_j) \qquad (6.6)$$

Para cada par  $p_j \leftrightarrow q_b$  del conjunto de posibles, se calcula la distancia entre el punto transformado  $G_r(p_j)$  y  $q_b$ . El minutiae  $q_b$  más cercano y que cumpla la condición  $|G_r(p_j) - q_b| \leq d_1$ , donde  $d_1$  es el límite de aceptación para el par, es un par coincidente.

La transformación de un minutiae  $p_j$  en  $G_r(p_j)$  se realiza con la siguientes fórmulas

$$G_r(p_j)_x = t_x + (s \cdot \cos\theta \quad -s \cdot \operatorname{sen}\theta) \begin{pmatrix} x_{p_j} \\ y_{p_j} \end{pmatrix} \quad (6.7)$$

$$G_r(p_j)_y = t_y + (s \cdot \operatorname{sen}\theta \quad s \cdot \cos\theta) \begin{pmatrix} x_{p_j} \\ y_{p_j} \end{pmatrix} \quad (6.8)$$

Donde  $t_x$  y  $t_y$  indican la translación en las coordenadas x e y del minutiae. Los parámetros  $r = (t_x, t_y, s, \theta)^t$  son los parámetros de la transformación y pueden ser calculados como sigue:

$$r = \frac{1}{\det} \begin{pmatrix} l_A & 0 & -\mu_{x_A} & \mu_{y_A} \\ 0 & l_A & -\mu_{y_A} & -\mu_{x_A} \\ -\mu_{x_A} & -\mu_{y_A} & k & 0 \\ \mu_{y_A} & -\mu_{x_A} & 0 & k \end{pmatrix} \begin{pmatrix} \mu_{x_B} \\ \mu_{y_B} \\ l_{A+B} \\ l_{A-B} \end{pmatrix} \quad (6.9)$$

Donde:

$$\mu_{x_A} = \sum_i^k x_{a_i}, \quad \mu_{x_B} = \sum_i^k x_{b_i} \quad (6.10)$$

$$\mu_{y_A} = \sum_i^k y_{a_i}, \quad \mu_{y_B} = \sum_i^k y_{b_i} \quad (6.11)$$

$$l_{A+B} = \sum_i^k (x_{a_i}x_{b_i} + y_{a_i}y_{b_i}) \quad (6.12)$$

$$l_{A-B} = \sum_i^k (x_{a_i}x_{b_i} - y_{a_i}y_{b_i}) \quad (6.13)$$

$$l_A = \sum_i^k (x_{a_i}^2 + y_{a_i}^2) \quad (6.14)$$

y

$$\det = k \cdot l_A - \mu_{x_A}^2 - \mu_{y_A}^2 \quad (6.15)$$

Donde  $k$  es el tamaño del vector de coincidentes.

El último paso del proceso de búsqueda de pares coincidentes consiste en la búsqueda del minutiae  $q_b$  más cercano a  $G_r(p_j)$  y que cumpla nuevamente la condición  $|G_r(p_j) - q_b| < d_1$ , para cada minutiae  $p_j$  restante en  $P$ . Si satisface la condición, entonces el par  $p_j \leftrightarrow q_b$  es tomado como par coincidente.

### 6.3 Verificación de las huellas dactilares

El criterio escogido en este trabajo para aceptar, o rechazar, una huella como del mismo individuo consiste en comprobar si al menos la mitad del conjunto más numeroso de minutias de las dos huellas ha sido emparejado. En ese caso, esa huella se acepta como del mismo individuo y se rechaza en caso contrario.

Este algoritmo ha sido probado con un total de dieciséis huellas dactilares, de las que se tenían ocho muestras distintas de cada una. El resultado ha sido que, un 65.5 % de las huellas dactilares que eran del mismo individuo han sido aceptadas como tales, y un 99.98% de las huellas que no eran de la misma persona han sido rechazadas. El porcentaje tan bajo de aceptación se debe a que varias de las muestras con las que se ha probado este algoritmo tenían mala calidad, lo que evitaba que los minutias fueran reconocidos con éxito. Sin embargo, todas las huellas de media o alta calidad han sido aceptadas por el algoritmo.



## Capítulo 7

# Conclusiones

En este proyecto, se ha realizado un trabajo de recolección de información sobre el campo de las huellas dactilares y la seguridad biométrica, ambos prácticamente desconocidos para mí al comienzo de este trabajo. Tras obtener una visión general del mundo de las huellas dactilares, se investigó el proceso de reconocimiento de huellas dactilares empleado en los sistemas de autenticación actuales, los cuales empleaban un procesamiento de la imagen y de la señal, Redes Neuronales Artificiales para extraer los minutiaes y, finalmente, un algoritmo para la comparación de los dos patrones.

Dada la restricción de tiempo, el trabajo inicialmente no se pensó como un AFIS completo. Pese a ello, y tras diversas modificaciones de la aplicación, se consiguió alcanzar ese objetivo, que cumple con todos los casos de uso descritos en el capítulo 2. Aunque el resultado es mejor de lo que se pensó en un principio, ha habido diversos fallos en la ejecución del proyecto, como por ejemplo la planificación temporal, que ha supuesto bastante más de lo previsto. Otro de los fallos cometidos ha sido la mala elección de algunas herramientas, como el gestor de bases de datos, que resultó ser poco eficiente. Finalmente, otro error fue elegir un filtro gaussiano frente a un filtro Gabor, dado que aplicar el filtro gaussiano es uno de los cuellos de botella de la aplicación. Esto no ocurriría en el caso de los filtros de Gabor, que son más eficaces y eficientes; sin embargo, su implementación es más compleja y hubiera aumentado el notablemente el tiempo invertido.

Aunque parte de la información necesaria para el trabajo ha sido recopilada de fuentes externas, gran parte de ella ha sido adquirida a lo largo de la carrera en diversas asignaturas como Matemáticas, todas las materias de Diseño de Software y Orientación a Objetos, cuyos conocimientos han servido, para que todas las modificaciones hechas a la aplicación a lo largo del trabajo no requirieran una gran tarea de modificación de muchas clases, sino que consistieran en pequeños cambios de alguna clase ya existente. En otras ocasiones, ha bastado simplemente con añadir más clases, cuya integración fue prácticamente

inmediata. Se han empleado los conocimientos adquiridos en las asignaturas de Inteligencia Artificial, Ingeniería del Conocimiento y Técnicas de Aprendizaje Automático en el desarrollo del capítulo 5 y la red neuronal. Se han usado los conceptos aprendidos en la asignatura de Computación Paralela para la paralelización de parte del código. Igualmente, se han aplicado nociones adquiridas en Diseño de Bases de Datos y conceptos de Sistemas Operativos para el instalador y las dependencias existentes entre Visual C++ y Windows.

A nivel personal, este trabajo me ha supuesto un reto, tanto por su dificultad como por su extensión. Con él he conseguido entender en mayor profundidad el funcionamiento de las Redes Neuronales Artificiales, el Aprendizaje Automático, el Procesamiento de Imágenes y la Visión por Ordenador.

## 7.1 Trabajo futuro

Las sugerencias adicionales para mejorar el funcionamiento de este trabajo:

- Aumento del rendimiento mediante la implementación del paralelismo con CUDA en vez de con OpenMP.
- El proceso de binarización puede ser mejorado mediante el reconocimiento automático de los parámetros del filtro, en vez de ser estáticos. Esto permitiría aumentar la calidad de las imágenes binarizadas obtenidas. En este sentido, si los filtros gaussianos fueran sustituidos por los de Gabor, aparte de obtener unas huellas de mayor calidad, se incrementaría el rendimiento notablemente.
- Incorporación de otros métodos biométricos distintos a la huella, como por ejemplo el iris o la oreja.
- Generar contraseñas únicas a partir de los minutiaes obtenidos de las huellas por el sistema para su uso en el acceso a sitios web de manera más segura, dado que las contraseñas generadas serían aleatorias y con más caracteres de lo habitual en las nuestras, y sin necesidad de recordarlas, ya que sería el propio programa quien las obtendría a partir de la huella.
- Mejora del algoritmo de comparación de huellas dactilares por uno invariante al número de minutiaes extraídos, o la sustitución de éste por una red neuronal capaz de reconocer el patrón de cada individuo.
- Integrar más modelos de escáneres de huellas dactilares.

# Bibliografía

- [1] John Edgar Hoover. *The science of fingerprints*. United states department of justice, FBI.
- [2] Anil K. Jain. *Uniqueness of Fingerprints*. Michigan State University.
- [3] Haykin. S. *Neural Networks and Learning Machines*. Prentice Hall, 2009.
- [4] SHIH-HSU CHANG y FANG-HSUAN CHENG y WEN-HSING HSU y GUO-ZUA WU. *Fast algorithm for point pattern matching: invariant to translations, rotations and scale changes*. Chung-Hua Polytechnic Institute Taiwan.
- [5] N.P. Khanyile y J.R. Tapamo y E. Dube. *A Comparative Study of Fingerprint Thinning Algorithms*. University of KwaZulu-Natal.
- [6] O’Gorman L. y Nickerson J.V. *An approach to fingerprint filter design*. AT&T.
- [7] O’Gorman L. y Nickerson J.V. *Matched Filter Design for Fingerprint Image Enhancement*. AT&T.
- [8] Xudong Jiang y Wei Yun Yau y Wee Ser. *Fingerprint Image Processing for Automatic Verification*. Nanyang Technological University.





## Capítulo 8

# Anexo I: manual de instalación

Para instalar la aplicación, es requisito indispensable contar con un ordenador con Windows 10 instalado. En el ordenador, deberemos ejecutar como administrador el instalador:

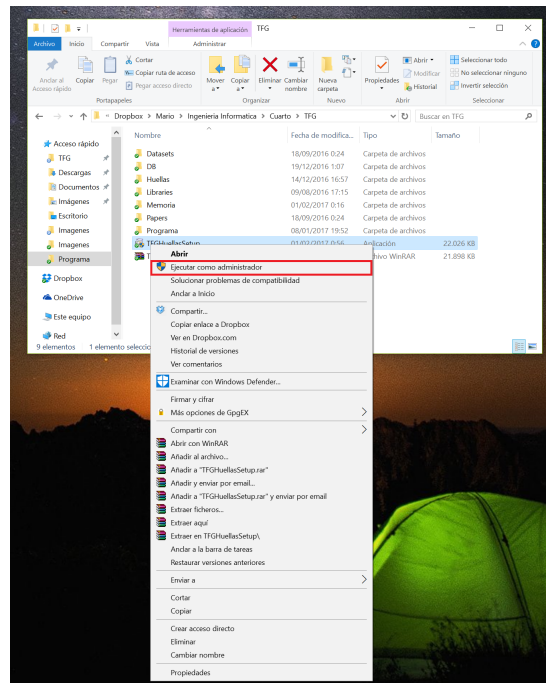


Figura 8.1: Instalador de la aplicación.

Este creará una carpeta llamada *TfgMarioRodriguez* en *C:/Program Files/* con el contenido de la aplicación y un acceso directo en el escritorio. Adicionalmente, se intentará instalar *Microsoft Visual C++ 2015 Redistributable*, dado que es necesario para poder ejecutar la aplicación; si ya lo tiene instalado en su ordenador, no es necesario que lo instale. En ocasiones, si decide instalar el *Microsoft Visual C++ 2015 Redistributable*, y ya lo tiene instalado en el ordenador, este mostrará un error, pero la instalación proseguirá normalmente.

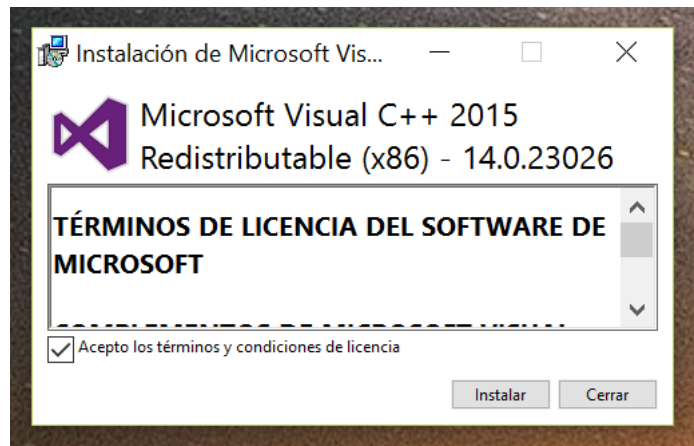


Figura 8.2: Instalación de Visual C++.

## Capítulo 9

# Anexo II: manual de usuario

Al abrir la aplicación de éste TFG, la primera pantalla que veremos será la siguiente:



Figura 9.1: Vista principal de la aplicación

Desde ella podemos acceder a diversas opciones, como la verificación de huellas dactilares, identificación de un individuo, su gestión, someter a un perceptrón al método Hold-Out o extraer los puntos de una, o varias huellas, a un archivo CSV.

## 9.1 Verificación de huellas dactilares

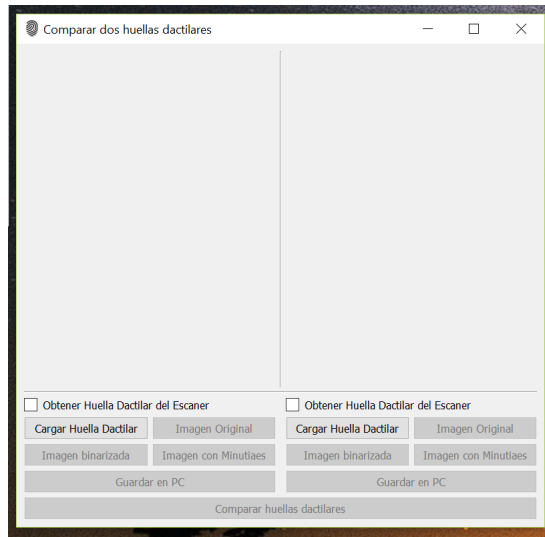


Figura 9.2: Vista principal de la verificación de huellas.

Para cargar una huella y poder compararla con otra ya cargada, o por cargar, se puede elegir entre dos métodos: cargarla desde una imagen u obtenerla desde un escáner de huellas dactilares. Si escogemos la opción del fichero de imagen, nos saldrá un cuadro de diálogo como en la figura siguiente:

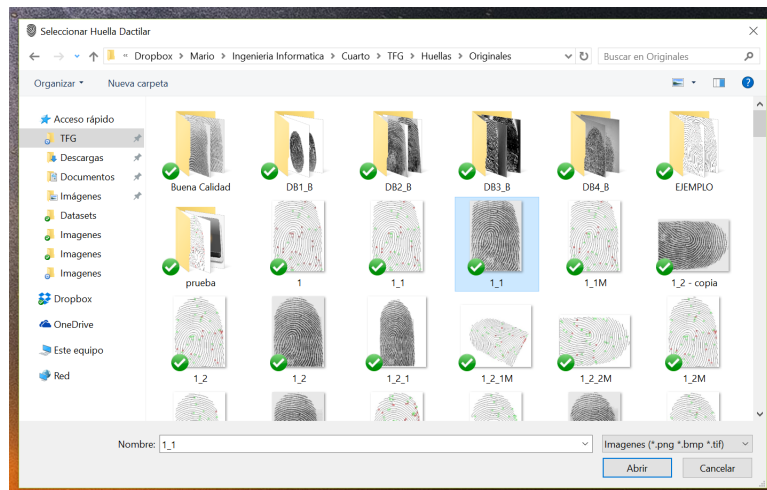


Figura 9.3: Vista al seleccionar la huella dactilar

En él cual podremos seleccionar la huella dactilar deseada, tras lo cual el sistema la cargará y mostrará por pantalla. Otra opción es capturar la huella dactilar desde el escáner; para ello, debemos conectarlo al ordenador y pulsar sobre el checkbox: *Obtener Huella Dactilar del Escáner*. Después de esto, ponemos el dedo sobre el escáner y buscamos la posición con la cual la huella dactilar tenga la mejor calidad posible.

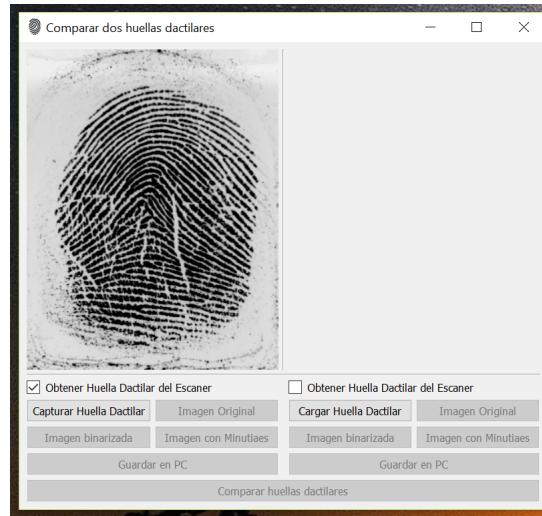
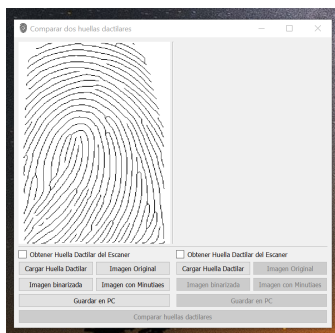


Figura 9.4: Vista al obtener una huella dactilar desde el escáner.

Una vez encontremos la posición deseada, debemos pulsar el botón *Capturar Huella Dactilar* y, al igual que al cargarla desde un fichero, el sistema cargará la imagen obtenida desde el escáner y la mostrará por pantalla. Después de esto, podemos ver la imagen original, binarizada y con los minutiaes extraídos de la huella dando a sus respectivos botones.



(a) Vista de la huella dactilar binarizada.



(b) Vista de la huella dactilar con los minutiaes resaltados.

Adicionalmente, podemos guardar todas sus imágenes en nuestro ordenador clicando en el botón: *Guardar en PC*. Tras pulsarlo, se nos abrirá un cuadro para seleccionar la carpeta donde deseamos guardarlas como el de la imagen siguiente:

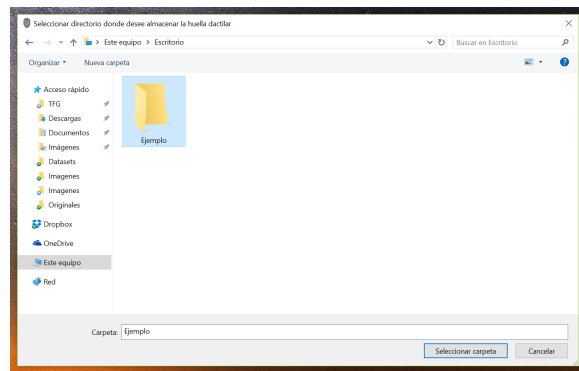


Figura 9.6: Vista para guardar la huella dactilar y sus imágenes.

Cuando hayamos seleccionado la carpeta, se nos guardarán las imágenes tal y como vemos en la imagen:

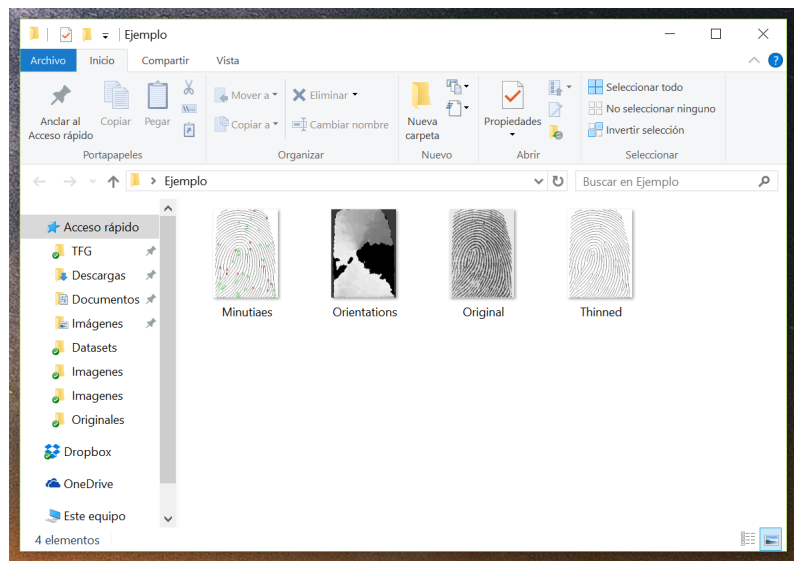


Figura 9.7: Vista del resultado de guardar las imágenes en una carpeta.

Una vez cargadas las dos huellas dactilares, se nos desbloqueará el botón de compararlas. Si lo pulsamos, el sistema las comparará y nos mostrará los resultados en una nueva ventana como la siguiente:

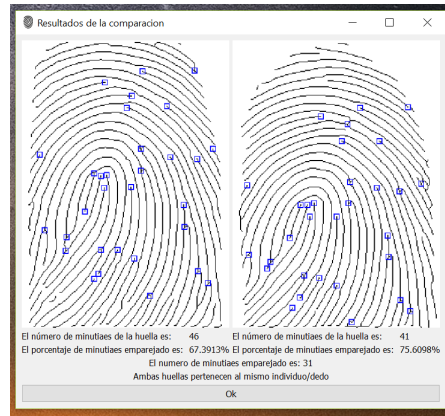


Figura 9.8: Vista del resultado de la verificación de dos huellas dactilares.

En esta ventana, podemos ver los minutias emparejados entre ambas resaltados en azul, así como el número de ellos en cada huella, el tanto por ciento emparejado sobre cada huella, el número de emparejados y si ambas huellas pertenecen, o no, al mismo individuo.

## 9.2 Identificación de un individuo

Esta vista es muy similar a la de verificación de huellas dactilares, y su funcionamiento, también. Primero, debemos cargar una huella dactilar, o tomarla desde el escáner, y después dar al botón de *Identificar Individuo*.

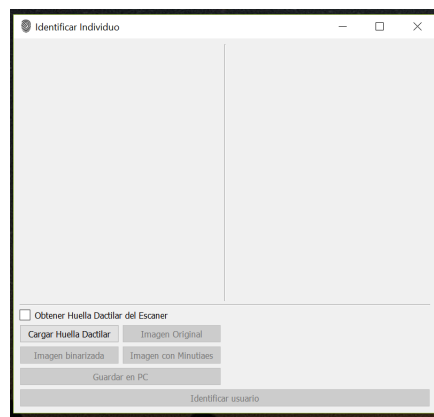


Figura 9.9: Vista principal de la identificación de un individuo mediante su huella dactilar.

Tras esto, el sistema buscará al individuo, si existiese, al que pertenece la huella cargada y mostrará los datos de este en un cuadro de diálogo, o una advertencia en caso de no encontrar ningún individuo coincidente.

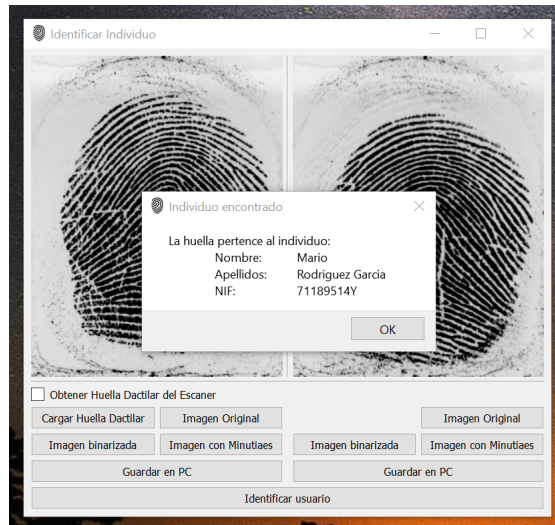


Figura 9.10: Vista del resultado de la identificación de un individuo.

### 9.3 Gestión de individuos

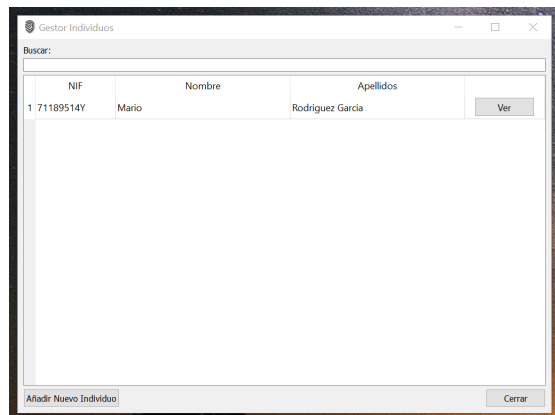


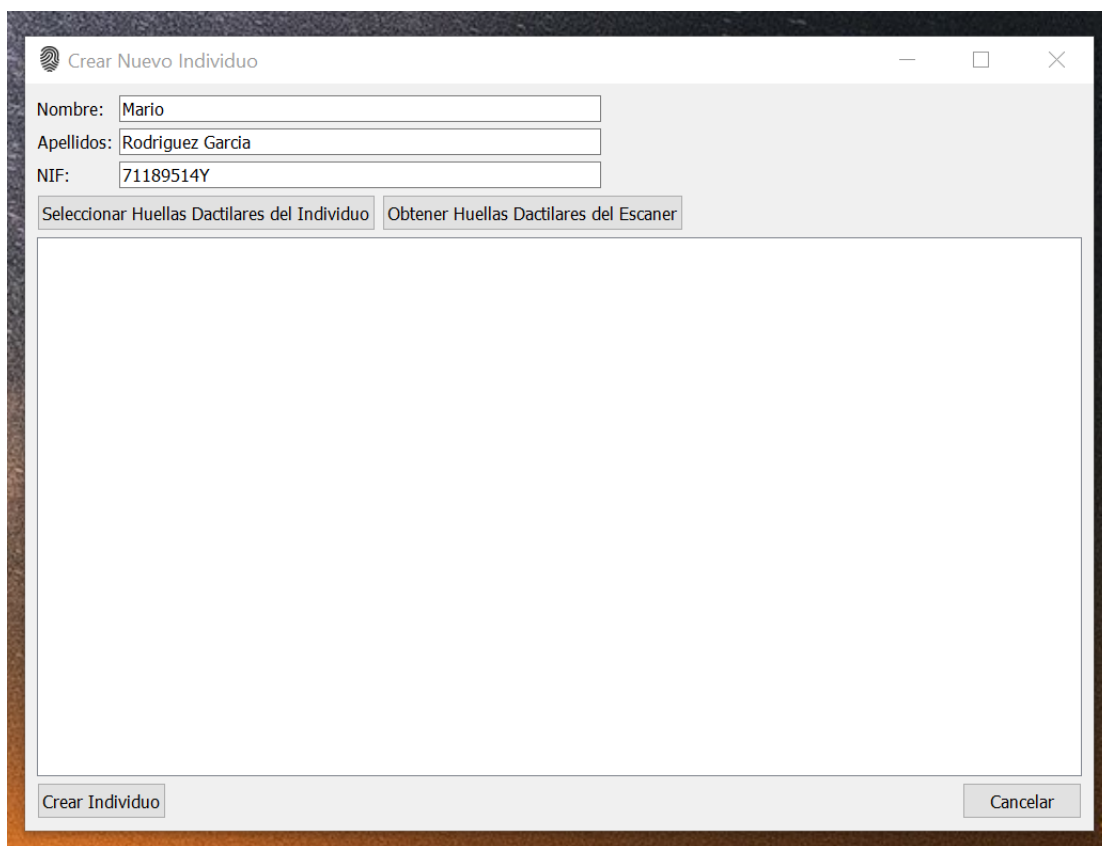
Figura 9.11: Vista principal de la gestión de individuos

En esta ventana, podremos gestionar los individuos, es decir, ver los detalles de un individuo o añadir nuevos individuos al sistema. Posee una línea de búsqueda que examina en todos los campos el texto introducido en ella.



## 9.4 Creación de un individuo

A la hora de crear un individuo, son obligatorios los campos de nombre, apellidos y NIF. Las huellas dactilares se pueden cargar con los métodos anteriormente vistos: cargarlas desde un fichero o desde un escáner.



The screenshot shows a window titled "Crear Nuevo Individuo" with a standard Windows-style title bar (minimize, maximize, close). The window contains a form with the following elements:

- Form fields:
  - Nombre: Mario
  - Apellidos: Rodriguez Garcia
  - NIF: 71189514Y
- Buttons below the form:
  - Seleccionar Huellas Dactilares del Individuo
  - Obtener Huellas Dactilares del Escaner
- Buttons at the bottom of the window:
  - Crear Individuo
  - Cancelar

Figura 9.12: Vista principal de la creación de un individuo.

Si escogemos cargar una huella dactilar desde el escáner, se nos abrirá una ventana que nos permitirá ver en cada momento la imagen tomada por el escáner para que podamos seleccionar la más adecuada a nuestras necesidades. Una vez hayamos decidido que la imagen es la que queremos, pulsamos en *Capturar Huella Dactilar* y esta se añadirá a las huellas a añadir al nuevo individuo.

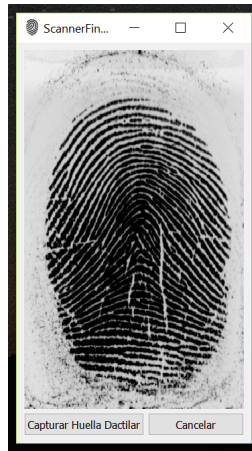


Figura 9.13: Vista de la captura de huellas dactilares para un individuo.

Si deseamos eliminar alguna de las huellas seleccionadas para cargar en el individuo, deberemos hacer clic sobre ella y pulsar la tecla *Suprimir*. Finalmente, cuando hayamos rellenado los datos correctamente y hayamos cargado todas las huellas deseadas, el siguiente paso es dar al botón de *Crear Individuo*. Tras esto, se procesarán todas las huellas y compararán con las ya existentes para evitar duplicados en individuos y se creará el nuevo individuo.

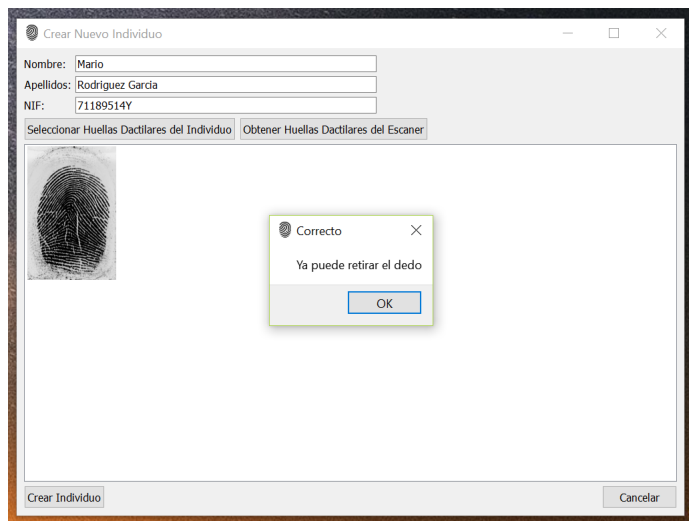


Figura 9.14: Vista del resultado de capturar una huella dactilar para un individuo.

## 9.5 Ver detalles de individuo

En esta pantalla, seremos capaces de añadir nuevas huellas dactilares al individuo, ver todas las imágenes de las huellas ya presentes en él, eliminar alguna de sus huellas o borrar al individuo del sistema.



Figura 9.15: Vista principal de los detalles de un individuo, así como de sus huellas dactilares.

Para ver los detalles y las imágenes de una huella dactilar, deberemos clicar sobre ella y el sistema mostrará a la derecha su imagen original, que podremos cambiar pulsando sobre los botones correspondientes. Por el contrario, si lo que deseamos es eliminar una huella, deberemos seleccionarla con el ratón y pulsar la tecla *Suprimir*. Tras lo cual, nos saldrá un cuadro de diálogo pidiendo la confirmación de la acción.

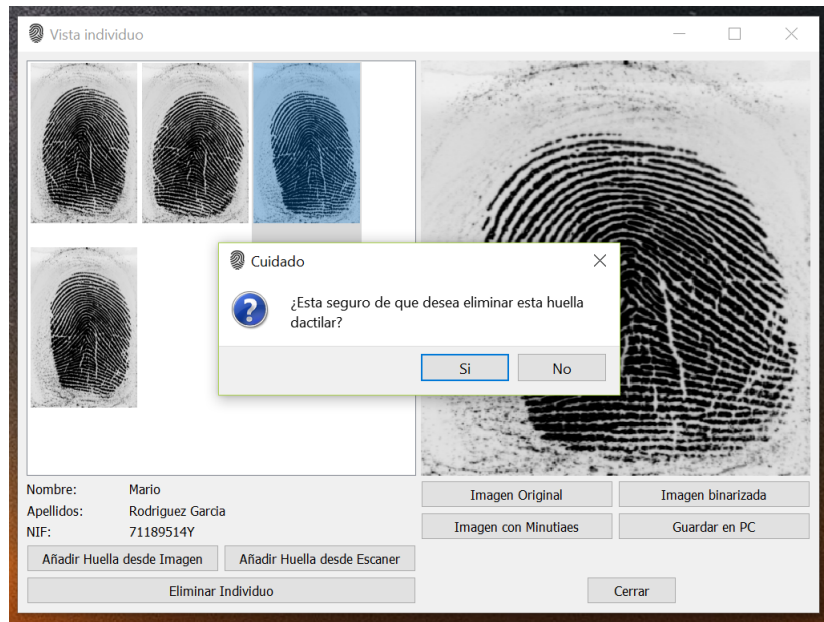


Figura 9.16: Ejemplo de eliminación de una huella dactilar.

Finalmente, si queremos eliminar el individuo del sistema, deberemos pulsar en el botón *Eliminar Individuo*, el cual nos pedirá una confirmación de la acción; tras aceptarla, lo borrará del sistema.

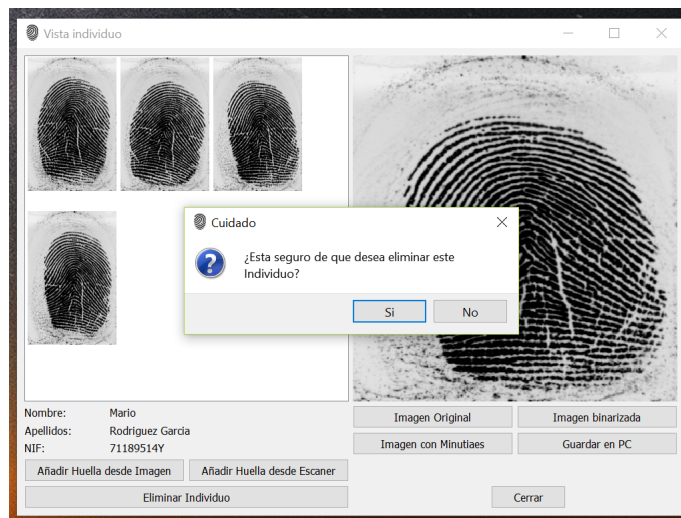


Figura 9.17: Ejemplo de eliminación de un individuo.

## 9.6 Comprobar eficiencia de un perceptrón

En esta ventana, vamos a ser capaces de evaluar la eficiencia de un perceptrón multicapa aplicando un método Hold-Out.

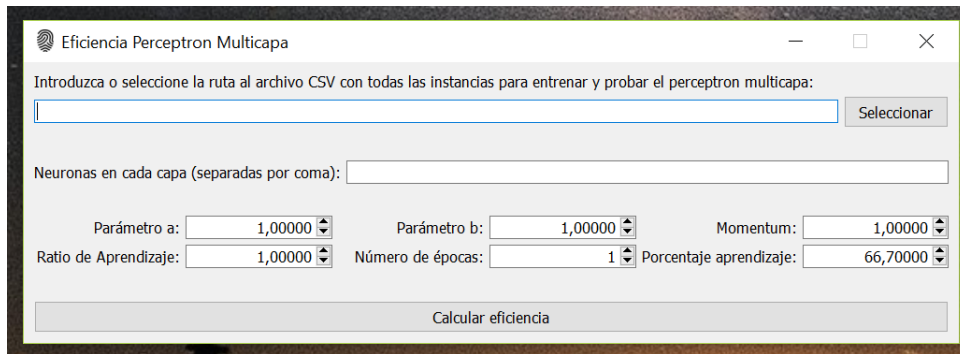


Figura 9.18: Vista principal del método Hold-Out para el perceptrón.

Lo primero de todo será seleccionar el conjunto de datos; para ello, escribimos su ruta en la línea o pulsamos el botón *Seleccionar* y elegimos el archivo en nuestro ordenador.

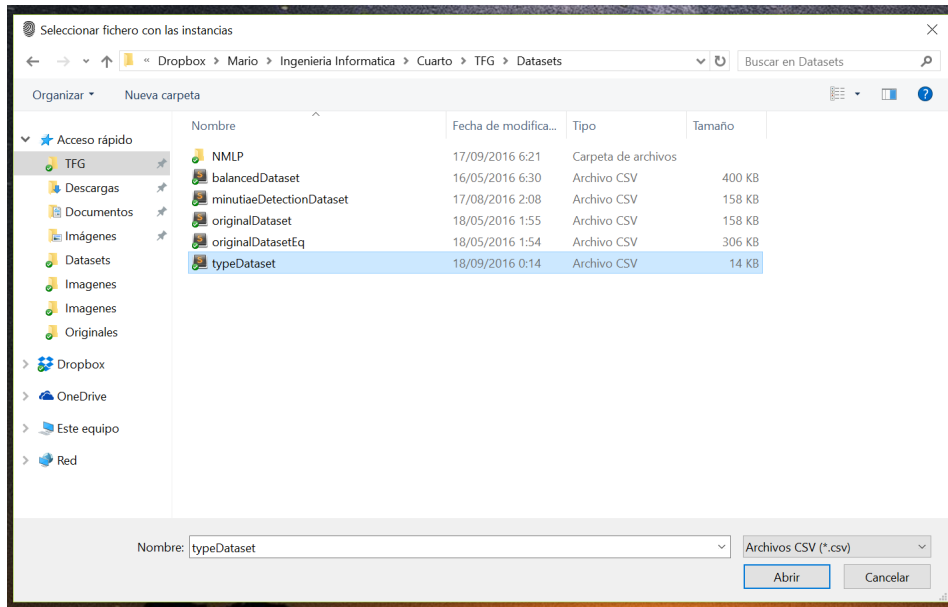


Figura 9.19: Vista de selección del conjunto de datos para el aprendizaje y prueba del perceptrón.

Tras haber seleccionado el archivo, deberemos introducir el número de neuronas del perceptrón en cada capa separadas por una coma. No es necesario introducir el número de entradas ni de clases, dado que eso lo detecta automáticamente el programa. El siguiente paso es ajustar los posibles parámetros del perceptrón, así como los del experimento Hold-Out.

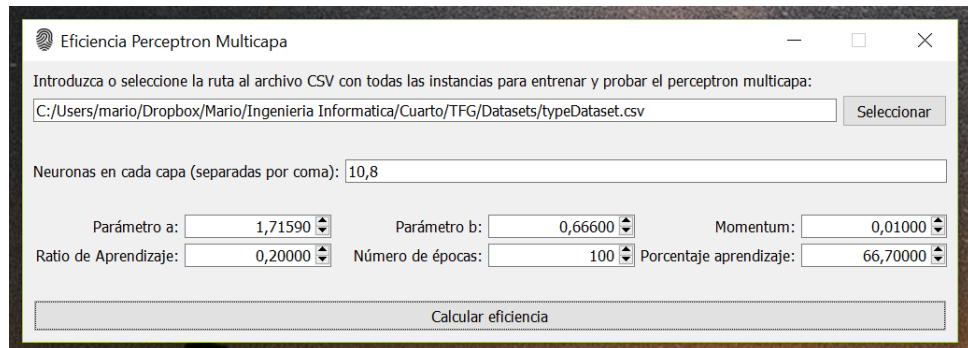


Figura 9.20: Ejemplo de ventana con todos los parámetros ajustados.

Cuando hayamos rellenado todos los datos necesarios para el experimento, deberemos hacer clic sobre el botón *Calcular Eficiencia*. Esto creará un cuadro de diálogo con la matriz de confusión resultante del experimento:

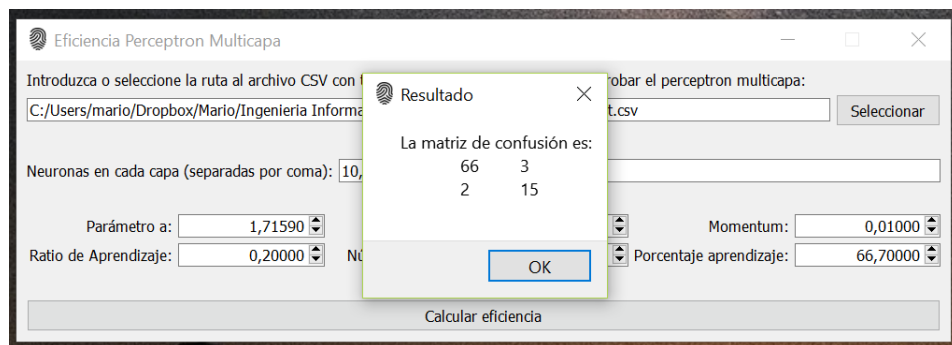


Figura 9.21: Vista del resultado del método Hold-Out.

## 9.7 Obtener minutiaes y minutiaes no repetidos

En ésta pantalla, se nos permitirá obtener los minutiaes, y no minutiaes, marcados sobre una huella dactilar binarizada. Los finales de línea han de marcarse con el color (34,177,76), y las bifurcaciones con (237,28,26); ambos colores son el verde y el rojo por defecto en Paint. Una vez marcados todos los minutiaes, vamos a las opciones del menú principal para extraerlos, que nos

permitirán seleccionar la carpeta donde esté la huella, o huellas en el caso de que los deseemos sin repetir, para extraerlos en esa misma carpeta en un archivo CSV.

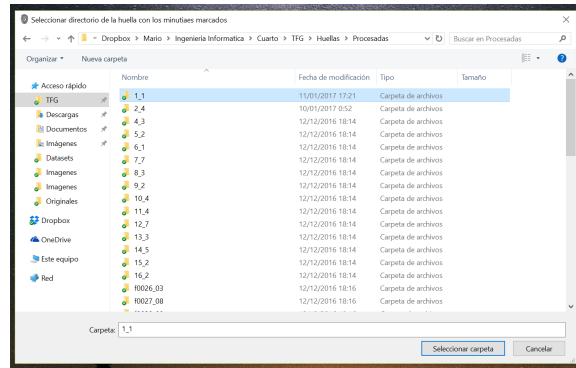


Figura 9.22: Vista de la selección de la carpeta para la extracción de minutiaes de una huella.

En el caso de varias huellas, el proceso tarda un rato, debido al gran número de comparaciones que debe realizar.



Figura 9.23: Vista del resultado de la extracción de minutiaes sobre una huella.





## Capítulo 10

# Anexo III: manual de despliegue en *Visual Studio Community 2015*

Para desplegar la aplicación en *Visual Studio Community 2015*, y que funcione correctamente, se deberá instalar la versión de *Qt 5.6* y su plugin para el entorno, ubicados en los siguientes links:

<https://download.qt.io/archive/qt/5.6/5.6.0/qt-opensource-windows-x86-msvc2015-5.6.0.exe>

[http://download.qt.io/official\\_releases/vsaddin/qt-vs-tools-msvc2015-2.1.1.vsix](http://download.qt.io/official_releases/vsaddin/qt-vs-tools-msvc2015-2.1.1.vsix)

Tras esto, tan solo deberemos abrir la *solution* bien desde el archivo *sln*, bien desde el propio *Visual Studio Community 2015* y seleccionar la opción *x86* para poder compilar y ejecutar la aplicación desde el código fuente.