



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Desarrollo de una aplicación de comparación de imágenes médicas

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

Autor: Raquel Jiménez Mondéjar

Tutor: Joan Andreu Sánchez Peiró

Curso 2019-2020

Agradecimientos

En primer lugar, me gustaría dar las gracias a mi tutor Joan Andreu por ayudarme y confiar en mí para llevar a cabo este proyecto.

También querría agradecer a Pedro y Elisa por su disposición y colaboración. Sin ellos, el desarrollo de este trabajo no habría sido posible. Gracias por abrirnos las puertas del Hospital Universitario Doctor Peset.

Y por último, gracias a mi familia y amigos, especialmente a mis padres, por ser un apoyo incondicional en todo momento pese a todas las circunstancias.

Resumen

Este proyecto se basa en la implementación de una aplicación con interfaz gráfica para realizar la comparación entre dos imágenes médicas correspondientes a un paciente antes y después de realizarle un tratamiento. La comparación de ambas imágenes es realizada a través del algoritmo de evaluación BIDM (Binary Image Distorsion Model). Este algoritmo realiza la comparación realizando un alineamiento no lineal minimizando la diferencia entre el píxel de la imagen a comparar con respecto al píxel de la imagen de referencia, teniendo en cuenta el valor de dos parámetros correspondientes al rango de distorsión y la ventana de contexto. En las pruebas realizadas se muestra la variación del resultado obtenido, dependiendo los valores asociados a cada parámetro con respecto a la imagen a comparar.

Los resultados obtenidos constan de un valor asociado a la similitud y una imagen correspondiente a la variación del nivel de gris entre ambas imágenes. Dicha variación ha sido interpretada con respecto a la imagen a comparar, teniendo en cuenta si las imágenes se corresponden a los estudios de medicina nuclear de gammagrafías óseas o DaTSCAN.

La aplicación ha sido desarrollada con la colaboración de los médicos del departamento de Medicina Nuclear del Hospital Universitario Doctor Peset, los cuales son usuarios finales de la propia aplicación.

Palabras clave: BIDM, comparación de imágenes, imagen médica, interfaz, C#

Abstract

This project is based on the implementation of a Graphical User Interface (GUI) application to compare two medical images of a patient before and after carrying out a treatment. The comparison of the two images is made by using the BIDM (Binary Image Distorsion Model) evaluation algorithm. This algorithm performs the comparison minimizing the difference between the pixel of the test image and the pixel of the reference image by a non-linear matching, taking into account the value of two parameters corresponding to the wrap range and context window. Tests done shows the variation of the results obtained depending on the value of each parameter according to the images of the comparison.

The results obtained contains a value associated with the similarity of both images and an image that shows the variation of grey level between them. Such variation has been interpreted taking into account the test image and if the type of image corresponds to the nuclear medicine studies of bone scan or DaTSCAN.

The application software has been developed in collaboration with physicians in the department of nuclear medicine at the Hospital Universitario Doctor Peset, which are end users of the application.

Key words: BIDM, image alignment, medical imaging, interface, C#

Índice general

Índice general	V
Índice de figuras	VII
Índice de tablas	VIII
<hr/>	
1 Introducción	1
1.1 Motivación	1
1.2 Objetivos	2
1.3 Impacto deseado	2
1.4 Metodología	2
1.5 Estructura de la memoria	3
2 Estado del arte	5
2.1 Imagen médica	5
2.2 Tecnologías en comparación de imágenes	6
3 Análisis del problema	9
3.1 Análisis de requisitos de la aplicación	9
3.1.1 Requisitos funcionales	9
3.1.2 Requisitos no funcionales	13
3.2 Análisis del marco legal y ético	13
3.3 Análisis de soluciones posibles y solución propuesta	13
4 Diseño de la solución	15
4.1 Diseño detallado	15
4.2 Tecnología utilizada	19
5 Desarrollo de la solución propuesta	23
5.1 Desarrollo de la interfaz	24
5.2 Desarrollo de la funcionalidad	26
5.2.1 Clase Programa	26
5.2.2 Clase Inicio	26
5.2.3 Clase Recorte	27
5.2.4 Clase Comparación	30
5.2.5 Clase Ajustes	35
5.2.6 Clase Imagen	35
6 Pruebas	37
6.1 Resultados variando las dimensiones del rectángulo de recorte y de la imagen	38
6.2 Resultados variando los parámetros del algoritmo	39
7 Conclusiones	43
Bibliografía	45
<hr/>	
Apéndice	
A Instalación de la aplicación	47

Índice de figuras

3.1	Diagrama de Casos de Uso	10
4.1	Prototipo ventana inicio	15
4.2	Prototipo ventana para cargar y recortar imágenes	16
4.3	Prototipo ventana para realizar la comparación	17
4.4	Prototipo ventana de ajustes del algoritmo de comparación	17
4.5	Diagrama de flujo de la aplicación	18
4.6	Algoritmo de evaluación Binary IDM (BIDM)	19
4.7	Parámetros del algoritmo de evaluación	20
4.8	Parámetros del algoritmo de evaluación	20
5.1	Solución ImagenCompara2	23
5.2	Ventana inicio de la aplicación	24
5.3	Ventana correspondiente a recortar imagen de la aplicación	24
5.4	Ventana correspondiente a realizar la comparación de la aplicación	25
5.5	Ventana para ajustar los parámetros del algoritmo de la aplicación	25
5.6	Clase Programa	26
5.7	Clase Inicio - Guardar ejecutables necesarios para la aplicación	27
5.8	CambiarRectángulo	28
5.9	Establecer punto de referencia - Paso 1	29
5.10	Establecer punto de referencia - Paso 2	29
5.11	Método <i>zoom</i>	30
5.12	Fragmento de código correspondiente a la conversión de la imagen de formato P5 a P2	31
5.13	Fragmento de código correspondiente a la llamada del algoritmo de comparación	31
5.14	Comparación gammagrafía osea 1	32
5.15	Comparación gammagrafía osea 2	33
5.16	Procedimiento para la obtención de la imagen interpretada	33
5.17	Comparación DaTSCAN 1	34
5.18	Comparación DaTSCAN 2	34
5.19	Clase Imagen	36
6.1	Paciente Anónimo	37
6.2	Comparación W=5 y C=5, imagen 100 %	38
6.3	Comparación W=5 y C=5, imagen 200 %	38
6.4	Comparación W=5 y C=5 de zona focalizada, imagen 200 %	39
6.5	Comparación W=1 y C=1 de zona focalizada, imagen 200 %	40
6.6	Comparación W=11 y C=11 de zona focalizada, imagen 200 %	40
6.7	Comparación W=2 y C=13 de zona focalizada, imagen 200 %	41
6.8	Comparación W=15 y C=5 de zona focalizada, imagen 200 %	41
6.9	Comparación W=2 y C=11 de zona columna focalizada, imagen 300 %	42
A.1	Carpeta ImagenCompara2	47

A.2	Ventana de instalación	47
-----	----------------------------------	----

Índice de tablas

3.1	Cargar imagen	10
3.2	Recortar imagen	11
3.3	Realizar zoom sobre imagen	11
3.5	Realizar comparación	11
3.4	Ajustar parámetros del algoritmo de comparación	12
3.6	Guardar imagen	12

CAPÍTULO 1

Introducción

El reconocimiento de imágenes es el área dentro de la inteligencia artificial que más avances ha tenido en los últimos años, especialmente gracias al desarrollo de nuevas tecnologías de aprendizaje profundo. En la actualidad existen diversos modelos capaces de reconocer y comparar imágenes de forma automática. En concreto, este área es muy compleja dado que para llevar a cabo la comparación entre dos imágenes se deben tener en cuenta múltiples factores, en su mayoría dependientes de las imágenes a procesar.

En concreto, en el ámbito médico, las imágenes se han vuelto casi imprescindibles para el diagnóstico y el tratamiento de pacientes. Debido a los avances tecnológicos en este ámbito, se permite obtener de forma no-invasiva, en la mayoría de los casos, con gran calidad de detalle la anatomía o información necesaria sobre cualquier aspecto funcional del paciente. Para realizar el diagnóstico existen multitud de tipos de imágenes según los estudios realizados sobre el paciente. En la mayoría de los casos, para realizar el análisis de dichas imágenes, el personal sanitario no cuenta con herramientas especializadas que les ayuden a realizar su diagnóstico. Debido a esto, el análisis realizado a través del uso de otra herramienta puede inducir a errores y zonas de la imagen que pueden pasar desapercibidas para el ojo humano, pero que podrían ser significativas para el diagnóstico, pueden no tenerse en cuenta.

En este trabajo, se pretende desarrollar una herramienta capaz de realizar un análisis entre dos imágenes médicas, en concreto sobre imágenes médicas obtenidas tras los estudios de medicina nuclear de gammagrafías óseas y DaTSCAN para determinar la evolución del paciente.

Dichas imágenes se corresponden con imágenes de un mismo paciente obtenidas antes y después de realizarle un tratamiento. Dependiendo del tipo de imagen a comparar correspondiente al estudio realizado, se pretende ajustar la comparación para que el resultado obtenido muestre la evolución correcta del paciente.

Este proyecto se ha llevado a cabo con la colaboración de médicos del área de medicina nuclear del Hospital Universitario Doctor Peset y aunque la aplicación se encuentra en fase de pruebas, se pretende dejar disponible una versión final para los médicos, dado que han mostrado gran interés en la misma.

1.1 Motivación

Como se ha comentado anteriormente, la comparación entre dos imágenes es una tarea compleja y laboriosa dado que se deben tener en cuenta factores propios de cada

par de imágenes que se quieren reconocer. Debido a esto, nace la necesidad de realizar una herramienta especializada en imágenes médicas capaz de comparar dos imágenes asociadas a un mismo paciente en distintos periodos de tiempo para poder determinar su correcta evolución y ayudar al personal sanitario con el diagnóstico a realizar.

De esta forma, debido a que actualmente prima el diagnóstico por imágenes, en pocos segundos con la aplicación desarrollada se puede ayudar al personal sanitario a consolidar su diagnóstico y a detectar posibles áreas que habrían pasado desapercibidas por los métodos de análisis anteriores pero que pueden presentar cierta relevancia en el diagnóstico de la evolución del paciente realizando así un diagnóstico precoz. Además de poder realizar un análisis focalizado sobre un área concreta.

También, esta herramienta está enfocada a la comparación de imágenes médicas dentro del área de la medicina nuclear y dado que hoy en día, existe un gran número de casos de pacientes que padecen una patología asociada a dicho área, cualquier ayuda al personal sanitario especializado o mejora en este ámbito, es bien recibida.

1.2 Objetivos

Los objetivos principales son:

- Crear una aplicación gráfica amigable para comparar imágenes médicas
- Desarrollar y estudiar los métodos automáticos de comparación de imágenes
- Aplicar técnicas eficientes para el análisis de imágenes 2D.
- Implementar una herramienta que obtenga resultados acordes a la evolución real del paciente.
- Ayudar al personal sanitario a interpretar el resultado de la comparación.

1.3 Impacto deseado

Actualmente no existe una herramienta especializada capaz de comparar imágenes médicas o zonas de interés de dichas imágenes correspondientes a los estudios de gammagrafías óseas o DaTSCAN y obtener una interpretación de las diferencias obtenidas correspondientes a la evolución del paciente. Por tanto, con esta aplicación se pretende suplir esa necesidad.

A día de hoy, la aplicación de este trabajo se encuentra en fase de pruebas. Pero se espera tener una versión estable apta para su uso desde cualquier centro médico.

Para ello, se desea que tanto el usuario final, correspondiente a cualquier personal sanitario, como todos los pacientes de dicha área puedan ejecutar nuestra aplicación o autorizar sus imágenes obtenidas para poder obtener las diferencias tras el tratamiento realizado y así, ayudar a asegurar el diagnóstico e identificar posibles anomalías que a simple vista no son percibidas.

1.4 Metodología

La realización de nuestra aplicación se ha llevado a cabo con una metodología ágil, puesto que a través de la retroalimentación recibida a lo largo de su implementación, se

han incorporado las mejoras y funcionalidades nuevas necesarias, de forma rápida y sin necesidad de rediseñar la aplicación.

Cabe destacar que la retroalimentación ha sido realizada por usuarios finales correspondientes a médicos del departamento de Medicina Nuclear del Hospital Universitario Doctor Peset, los cuales nos han facilitado las imágenes médicas necesarias para realizar las distintas pruebas sobre la aplicación, además de explicarnos cómo se debe realizar el correcto diagnóstico según el tipo de imágenes correspondientes a cada estudio realizado.

Para llevar a cabo la implementación de la aplicación, en primer lugar, se realizó una reunión con dichos usuarios para establecer las funcionalidades básicas necesarias que debían constar en la herramienta. Tras la implementación de dichas funcionalidades se realizaron diversas reuniones asociadas a distintas versiones de la aplicación para evaluar su correcto funcionamiento y detectar nuevas funcionalidades no implementadas en la versión anterior.

Dichas funcionalidades principalmente fueron asociadas a cambios focalizados en algún aspecto concreto sobre alguna funcionalidad ya implementada en la versión anterior tales como mejoras para facilitar el uso de la interfaz para que fuese más intuitiva para el usuario y modificaciones para realizar una interpretación más ajustada a la evolución real del paciente según los procedimientos que sigue el personal médico para realizar el diagnóstico correspondiente.

1.5 Estructura de la memoria

Esta memoria consta de siete capítulos. En el primer capítulo se realiza una pequeña introducción sobre el tema relativo del proyecto.

En el segundo capítulo se trata el estado del arte actual de las imágenes médicas, en concreto con los tipos de imágenes médicas existentes asociadas a los estudios de medicina nuclear, y de las tecnologías existentes en comparación de imágenes con modelos que realicen un alineamiento no lineal, haciendo una pequeña comparativa entre ellos y justificando la elección de nuestra implementación.

En el tercer capítulo se realiza un breve análisis del problema a tratar y los aspectos que debemos tener en cuenta, tales como los requisitos, antes de realizar el desarrollo de nuestra aplicación.

En el cuarto capítulo se describe detalladamente el diseño de la aplicación y las funcionalidades a implementar teniendo en cuenta las limitaciones del usuario.

En el quinto capítulo se describe cómo se ha llevado a cabo el desarrollo de la aplicación, explicando las clases y métodos utilizados para programar la funcionalidad requerida.

En el sexto capítulo se muestran los resultados obtenidos sobre un caso particular de un paciente anónimo, variando los parámetros del algoritmo de comparación utilizado y las dimensiones de las imágenes a comparar para observar cómo dichos parámetros afectan al resultado obtenido.

Y por último, en el séptimo capítulo se realiza una breve conclusión sobre el funcionamiento de la aplicación y los parámetros óptimos para obtener un mejor resultado según las imágenes a comparar junto con futuras ampliaciones.

CAPÍTULO 2

Estado del arte

En la actualidad, el área concerniente al reconocimiento y procesamiento de imágenes a través de procedimientos basados en inteligencia artificial se encuentra en pleno desarrollo.

En este capítulo se realiza un breve repaso sobre el estado del arte en imagen médica, especialmente en el área de medicina nuclear, además de los diferentes modelos en comparación de imágenes que no realicen una comparación lineal.

2.1 Imagen médica

La imagen médica es el conjunto de técnicas y procesos usados para crear imágenes del cuerpo humano, o partes de él, con propósitos clínicos o para la ciencia médica, según [10]. El tipo de imagen utilizada dependerá de los síntomas del paciente y de la parte del cuerpo que se quiere examinar. Las diferentes técnicas utilizadas son rayos X, tomografías computarizadas, estudios de medicina nuclear, imágenes por resonancia magnética y ecografías [11].

En concreto, los estudios de medicina nuclear utilizan pequeñas cantidades de materiales radiactivos llamados radiosondas que se inyectan, se inhalan o se tragan. Algunos de los estudios de medicina nuclear más comúnmente usados son [7, 6, 12]:

- **Gammagrafías óseas:** se estudia el sistema óseo desde el punto de vista anatómico y funcional para diagnosticar enfermedades óseas a través de la inyección de material radioactivo. La sustancia viaja a través de la sangre hasta los huesos y órganos. La cámara toma imágenes de cuánta cantidad de marcador radioactivo se acumula en los huesos. Las anomalías aparecen como zonas de hueso anormal que absorben más o menos cantidad del radiofármaco, que aparece más brillante o más oscuro que el hueso normal en la gammagrafía ósea.
- **Tomografía por emisión de positrones (PET) y Tomografía computarizada de emisión monofotónica (SPECT):** se emplean en el estudio del cáncer, del corazón y del cerebro. En ambas técnicas se detecta radiación gamma aunque entre ambas difiere el radiofármaco y la cámara utilizada, dado que en la SPECT el isótopo produce directamente el rayo gamma, mientras que en la PET el isótopo produce un positrón que es aniquilado con un electrón para producir los dos rayos gamma, según [19].
- **Gammagrafía tiroidea:** se suele emplear para encontrar cáncer de tiroides. El paciente ingiere yodo radioactivo que pasa al torrente sanguíneo y se acumula en la glándula de la tiroides.

- Exploraciones MUGA: se utiliza para evaluar la función cardíaca. Se usa para revisar la función del corazón, antes, durante y después de un tratamiento de quimioterapia para medir la cantidad de sangre bombeada por el corazón.
- Gammagrafías con galio: se emplea para identificar inflamación, infección o cáncer en el cuerpo utilizando galio como material radiactivo. El escáner busca los lugares donde dicho material se ha acumulado y dichas zonas pueden corresponder a inflamación o cáncer.
- DaTSCAN: se utiliza para detectar la pérdida de terminaciones nerviosas dopaminérgicas en el cuerpo estriado del cerebro a través de la inyección por vía intravenosa del material radiactivo Ioflupano.

2.2 Tecnologías en comparación de imágenes

En la literatura existen multitud de modelos de reconocimiento de imágenes, pero los principales modelos que realicen un alineamiento no lineal son 2DW (*Two-dimensional model*), P2DHMM (*Pseudo 2-Dimensional Hidden Markov Model*), P2DHMDM (*Pseudo 2-Dimensional Hidden Markov Model*) e IDM (*Image Distortion Model*), los cuales se encuentran explicados en el artículo [2].

En dicho artículo se detalla que los modelos pueden clasificarse en tres tipos: orden dos, orden uno y orden cero. Se dice que un modelo es de orden dos si el alineamiento de un píxel depende de los alineamientos realizados en sus píxeles vecinos en ambas direcciones en la cuadrícula de una imagen bidimensional. Un modelo es de orden uno si el alineamiento se realiza con respecto a sus vecinos en una dimensión y un modelo es de orden cero si no se tienen en cuenta la alineación de los vecinos adyacentes para realizar su propio alineamiento.

El modelo 2DM es un modelo de orden dos dado que alinea cada píxel teniendo en cuenta sus píxeles vecinos o adyacentes en ambas direcciones. Como restricciones, los píxeles adyacentes horizontalmente y verticalmente no pueden corresponderse a píxeles que se desvíen más de un píxel con respecto a la posición relativa en la imagen original. Para calcular la mejor alineación, el coste computacional es exponencial.

Por otra parte, el modelo P2DHMM es un modelo de orden uno dado que alinea cada píxel teniendo en cuenta los píxeles vecinos en una única dirección, los píxeles adyacentes horizontalmente. Para ello, el desplazamiento horizontal es el mismo para todos los píxeles de una columna y el desplazamiento vertical es cero.

Además, el modelo P2DHDHMDM también es un modelo de orden uno. Éste se corresponde con una implementación del modelo anterior, el modelo P2HMM, tras realizar una relajación con respecto al desplazamiento vertical de como máximo un píxel de diferencia.

Por último, el modelo IDM es un modelo de orden cero dado que no tiene en cuenta los píxeles adyacentes, si no otros parámetros como el rango de distorsión W y la ventana de contexto local.

En el artículo citado anteriormente al comienzo de esta sección, se han realizado diversas pruebas con ambos modelos para el reconocimiento de imágenes de distintas bases de datos. Los resultados obtenidos son que el modelo P2DHDHMDM obtiene una menor tasa de error que el resto de modelos, aunque el modelo IDM obtiene una tasa de error muy próxima a éste. Pero si se tiene en cuenta la complejidad computacional y la relación de ésta con la tasa de error, el mejor modelo se corresponde con el modelo IDM.

En la implementación de nuestra aplicación se utiliza el algoritmo BIDM, el cual utiliza dicho modelo para su implementación tras realizar un pequeño cambio que será explicado en la sección 4.2.

Dicho algoritmo se ha utilizado para estudiar el alineamiento de imágenes 2D en otras áreas tales como el reconocimiento automático de expresiones matemáticas [1]. Este área presenta diversos problemas dado que se debe realizar un correcto reconocimiento de los símbolos matemáticos y de la relación de éstos con respecto a la estructura de la expresión. Sin embargo, los resultados obtenidos en las pruebas realizadas en el artículo citado anteriormente muestran que este modelo consigue solventar los problemas anteriores e identificar correctamente las zonas de diferencias entre ambas aunque no puede determinar el origen de los errores. Debido a esto, se ha decidido utilizar el algoritmo BIDM para comparar las imágenes de nuestra aplicación.

CAPÍTULO 3

Análisis del problema

La aplicación a desarrollar en este trabajo consiste en una herramienta destinada para personal sanitario cuya funcionalidad principal es realizar una comparación entre dos imágenes de un mismo paciente referentes a distintos instantes de tiempo, y mostrar el resultado obtenido para que el personal sanitario pueda obtener sus conclusiones con respecto al diagnóstico del paciente. En la actualidad existe mucho interés con respecto a el ámbito referente a nuestra aplicación, el cual se encuentra en desarrollo.

Antes de llevar a cabo el desarrollo de nuestra aplicación se ha realizado un estudio sobre los requisitos y cualquier aspecto importante que se debe tener en cuenta, así como de la viabilidad de los mismos para poder ser implementados y cumplidos, respectivamente. Para ello, a continuación se encuentra el análisis de requisitos, y del marco legal y ético, además de la identificación y análisis de las posibles soluciones junto con la solución propuesta.

3.1 Análisis de requisitos de la aplicación

Para realizar el análisis de requisitos de nuestra aplicación, se realizó una reunión con los usuarios finales para identificar las funcionalidades necesarias a implementar y las posibles restricciones. A continuación, se encuentran detallados los requisitos funcionales y no funcionales de la aplicación.

3.1.1. Requisitos funcionales

Los requisitos funcionales de nuestra aplicación se encuentran detallados a través de un diagrama de casos de uso como ilustra la figura 3.1. En él se muestra que sólo existe un actor que interactúa con la aplicación, el cual corresponde con cualquier personal sanitario que desee utilizar la herramienta para ayudarlo con el diagnóstico del paciente, y las acciones que éste puede realizar para llevar a cabo la funcionalidad requerida. Cada caso de uso se encuentra explicado en las tablas concesivas.

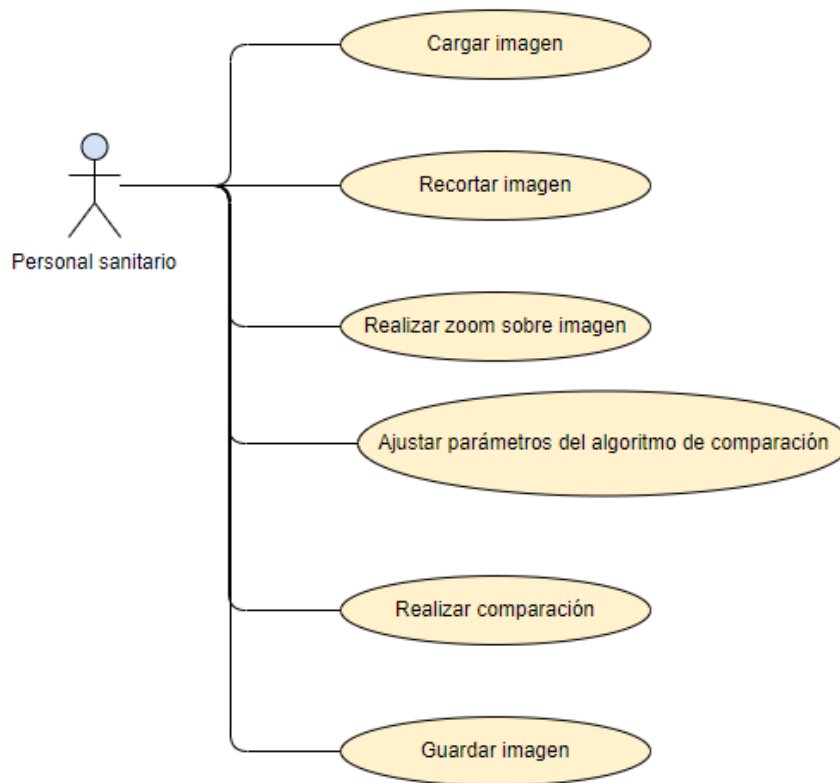


Figura 3.1: Diagrama de Casos de Uso

Caso de Uso	Cargar imagen
Actores	Personal sanitario
Resumen	Permite al usuario elegir y cargar cualquier imagen para realizar la comparación
Precondiciones	–
Postcondiciones	–
Incluye	–
Extiende	–
Hereda de	–
Flujo de eventos	
Actor	Sistema
1. El usuario selecciona la opción de cargar imagen	
	2. El sistema muestra el cuadro de dialogo para abrir la imagen
3. El usuario selecciona la imagen a cargar	
	4. La imagen seleccionada se muestra a través de la interfaz.

Tabla 3.1: Cargar imagen

Caso de Uso	Recortar imagen
Actores	Personal sanitario
Resumen	Permite al usuario seleccionar el área de la imagen que desea comparar.
Precondiciones	La imagen a recortar debe estar cargada.
Postcondiciones	–
Incluye	–
Extiende	–
Hereda de	–
Flujo de eventos	
Actor	Sistema
1. El usuario selecciona el área que desea comparar	
	2. El recorte del area seleccionada se muestra a través de la interfaz.

Tabla 3.2: Recortar imagen

Caso de Uso	Realizar zoom sobre imagen
Actores	Personal sanitario
Resumen	Permite al usuario redimensionar la imagen.
Precondiciones	La imagen a recortar debe estar cargada.
Postcondiciones	–
Incluye	–
Extiende	–
Hereda de	–
Flujo de eventos	
Actor	Sistema
1. El usuario cambia el valor asociado al zoom	
	2. La imagen se redimensiona.

Tabla 3.3: Realizar zoom sobre imagen

Caso de Uso	Realizar comparación
Actores	Personal sanitario
Resumen	Permite al usuario realizar la comparación de las dos imágenes recortadas y obtener el resultado de la comparación
Precondiciones	Las dos imágenes a comparar deben estar recortadas
Postcondiciones	–
Incluye	–
Extiende	–
Hereda de	–
Flujo de eventos	
Actor	Sistema
1. El usuario selecciona la opción de realizar comparación.	
	2. El sistema muestra el resultado obtenido a través de la interfaz.

Tabla 3.5: Realizar comparación

Caso de Uso	Ajustar parámetros del algoritmo de comparación
Actores	Personal sanitario
Resumen	Permite al usuario cambiar los parámetros del algoritmo utilizado para realizar la comparación para ajustar el resultado según las imágenes a comparar.
Precondiciones	–
Postcondiciones	–
Incluye	–
Extiende	–
Hereda de	–
Flujo de eventos	
Actor	Sistema
1. El usuario selecciona la opción de cambiar los parámetros del algoritmo.	
	2. El sistema abre una nueva ventana para ajustar dichos parámetros
3. Cambia los valores asociados a cada parámetro según sus preferencias.	
	4. Los nuevos valores son procesados y se comprueba su validez.

Tabla 3.4: Ajustar parámetros del algoritmo de comparación

Caso de Uso	Guardar imagen
Actores	Personal sanitario
Resumen	Permite al usuario guardar la imagen resultante tras realizar la comparación en su equipo.
Precondiciones	Se debe haber realizado la comparación entre dos imágenes
Postcondiciones	–
Incluye	–
Extiende	–
Hereda de	–
Flujo de eventos	
Actor	Sistema
1. El usuario selecciona la opción de guardar imagen.	
	2. El sistema muestra el cuadro de diálogo para guardar la imagen.
3. El usuario selecciona el directorio donde desea guardar la imagen	
4. El usuario establece el nombre de la imagen.	

Tabla 3.6: Guardar imagen

3.1.2. Requisitos no funcionales

El requisito no funcional principal es que la aplicación tiene que poder ejecutarse en los equipos del hospital, los cuales tienen Windows 7 con un sistema operativo de 32 bits. Por tanto, todos los ejecutables que se vayan a utilizar tienen que estar compilados para una arquitectura de 32 bits.

Además, el uso de la aplicación debe ser intuitivo y el tiempo de aprendizaje del usuario debe ser inferior a 3 horas. Por último, debido a que el personal sanitario normalmente tiene permisos restringidos sobre el equipo, la instalación y ejecución de la aplicación debe realizarse de forma sencilla y de acuerdo con los permisos del usuario.

3.2 Análisis del marco legal y ético

La aplicación sólo podrá ser utilizada por el personal sanitario autorizado tras recibir el consentimiento del paciente para tratar sus datos. Aunque nuestra aplicación no identifica al paciente y no se almacena ninguna información sobre el resultado obtenido ni sobre las imágenes médicas utilizadas que pueda ser relacionada con el paciente, la herramienta ha desarrollado en este trabajo si trata las imágenes para realizar la comparación.

Debido a esto se debe hacer un uso correcto de dichas imágenes y sólo las imágenes que han recibido el consentimiento del paciente para su utilización pueden ser comparadas. En las pruebas realizadas para comprobar el funcionamiento de la aplicación, se utilizarán imágenes médicas anónimas de pacientes del área de medicina nuclear del Hospital Universitario Doctor Peset que han autorizado su obtención y utilización.

3.3 Análisis de soluciones posibles y solución propuesta

Para llevar a cabo la elección del lenguaje de programación y el entorno de desarrollo a utilizar se han tenido en cuenta los siguientes aspectos. Para realizar una aplicación multiplataforma los lenguajes más utilizados y con mayor popularidad hoy en día son Java y Python. Pero la mayoría de programas desarrollados exclusivamente para Windows con entorno gráfico son desarrollados en los lenguajes C++ o C# junto con el entorno de trabajo .NET de Windows.

Por tanto, dado que nuestra aplicación va a ser utilizada exclusivamente en Windows, la solución propuesta en este trabajo ha sido desarrollada con el entorno de trabajo .NET de Windows a través del entorno de desarrollo Microsoft Visual Studio utilizando el lenguaje de programación C#.

CAPÍTULO 4

Diseño de la solución

En este capítulo, se procede a detallar el diseño realizado antes de llevar a cabo la implementación de nuestra solución y la tecnología utilizada para realizar la comparación. Debido a que nuestra aplicación no persiste ningún dato correspondiente al paciente ni al resultado asociado de la comparación, para realizar el desarrollo de esta herramienta no se requiere modelar ninguna entidad. Por tanto, el diseño se encuentra enfocado a la programación de la interfaz y las clases asociadas necesarias.

4.1 Diseño detallado

Para realizar el diseño de la interfaz, se ha realizado el diagrama de flujo de la aplicación representado en la figura 4.5 y se han prototipado las diferentes ventanas que forman parte de nuestra aplicación a través de la aplicación web Moqups [13]. A continuación, se encuentra detallado el diseño realizado siguiendo el flujo de la aplicación.

Tras iniciar la aplicación y pulsar el botón de inicio presente en la ventana correspondiente a la figura 4.1, aparece la ventana correspondiente a la figura 4.2. En dicha ventana, nuestra aplicación debe permitir al usuario cargar las imágenes que desea comparar de forma individualizada y que una vez éstas hayan sido seleccionadas y cargadas, se muestren a través de la interfaz. Puesto que puede darse el caso de que el usuario no quiera comparar toda la imagen sino un área de interés, la interfaz debe permitir al usuario poder recortar dicha área para poder realizar la comparación correspondiente.

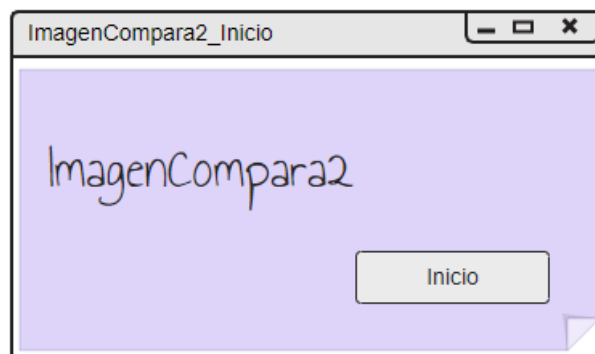


Figura 4.1: Prototipo ventana inicio

Para seleccionar el área de interés de cada imagen, a través de la interfaz se puede ajustar un rectángulo de recorte distinto para cada imagen. De esta forma, cada imagen puede ser recortada con dimensiones distintas según las necesidades del usuario. Por

defecto, aparecerá un rectángulo de recorte predeterminado sobre cada imagen cargada y a través de botones representando las cuatro acciones de subir, bajar, mover hacia la derecha y mover hacia la izquierda, correspondientes a cada imagen, el usuario puede desplazar el rectángulo de recorte sobre toda la imagen. Además, con el uso del ratón, el usuario también puede cambiar la posición del rectángulo de recorte. Si el usuario pulsa con el cursor sobre un punto fijo de la imagen, el rectángulo de recorte se mueve situándose el vértice superior derecho en el punto establecido por el usuario.

Si el usuario desea cambiar las dimensiones por defecto del rectángulo de recorte, a través de la interfaz se puede modificar su anchura y altura. Dichos parámetros se muestran a través de la interfaz y a través de botones para aumentar o disminuir su valor, o introduciendo el nuevo valor sobre un cuadro de texto, las dimensiones cambian. Además, también se puede modificar dichas dimensiones a través de la utilización del ratón. Si el usuario pulsa con el cursor sobre un punto de la imagen, arrastra sobre cualquier dirección manteniendo pulsado el ratón y suelta sobre otro punto distinto, el rectángulo de recorte, cambia sus dimensiones siendo el punto determinado al pulsar y el punto establecido por la posición del cursor al soltar el botón del ratón, vértices opuestos del rectángulo.

En el caso de que se quieran reajustar las dimensiones de la imagen para facilitar al usuario la selección del área a recortar, las dimensiones de la imagen se pueden agrandar o disminuir realizando *zoom* sobre ella a través de la interfaz cambiando su valor o a través de la rueda de desplazamiento. Además, el *zoom* realizado asociado a cada imagen será mostrado a través de la interfaz.

Si el usuario desea copiar las dimensiones del rectángulo de recorte asociado a una imagen en la otra imagen a comparar para establecer el mismo rectángulo, a través de un punto de referencia asociado a cada imagen, el usuario puede realizar dicha acción para facilitarse el ajuste de las áreas de las imágenes a comparar, en concreto para poder establecer dos imágenes que presenten las mismas dimensiones con facilidad. Esta funcionalidad no fue detectada en un análisis y diseño inicial, pero tras realizar las diversas reuniones con los usuarios finales y pruebas de versiones iniciales, se identificó este problema que se encuentra desarrollado en el capítulo siguiente aunque no aparece en los prototipos.

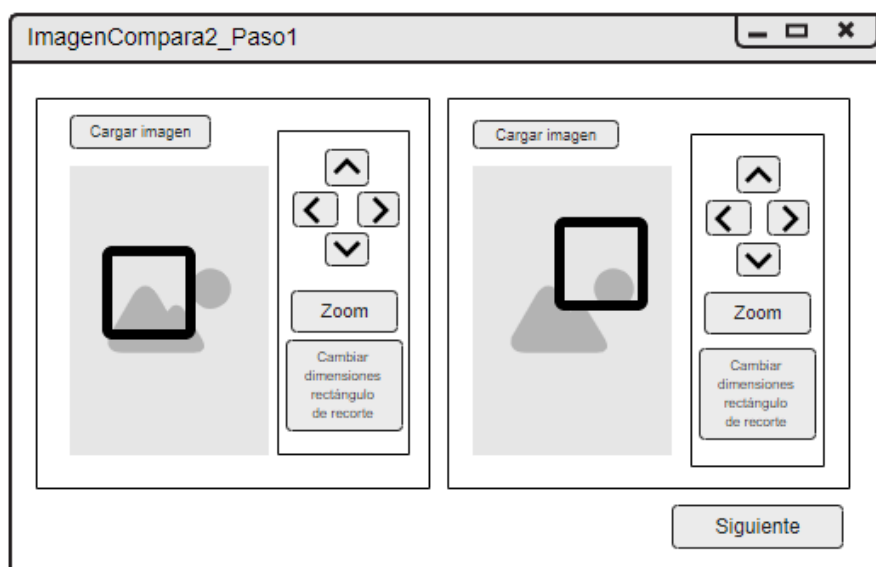


Figura 4.2: Prototipo ventana para cargar y recortar imágenes

Una vez se haya establecido el rectángulo de recorte en ambas imágenes según las áreas a comparar que el usuario quiera, a través de un botón de siguiente situado en la parte inferior, aparece la ventana de la figura 4.3 para llevar a cabo la comparación. En dicha ventana, aparecen las dos imágenes recortadas junto con el botón para realizar la comparación. Por defecto, el algoritmo de comparación, tiene unos valores predeterminados, que pueden ser cambiados a través de la opción del menú superior con la opción de ajustes cambiando los valores de la nueva ventana emergente de la figura 4.4.

Para llevar a cabo la comparación, el usuario debe pulsar el botón correspondiente a realizar comparación. Su evolución se muestra en la barra de progreso inferior y cuando ésta se encuentra completada, aparece el valor del resultado obtenido, la imagen resultante de la comparación y la imagen interpretada del resultado. Ambas imágenes se pueden guardar en el equipo para que el personal sanitario pueda registrar la evolución del paciente.

La comparación se realiza tomando como referencia la imagen situada más hacia la izquierda en la ventana, la cual debe corresponder a un periodo de tiempo anterior a la otra. Si se desea cambiar la referencia, el usuario puede pulsar el botón correspondiente a intercambiar imágenes. En ese caso, las imágenes intercambian su posición y si ya se hubiese realizado una comparación, todos los campos correspondientes al resultado de la comparación se reinician.

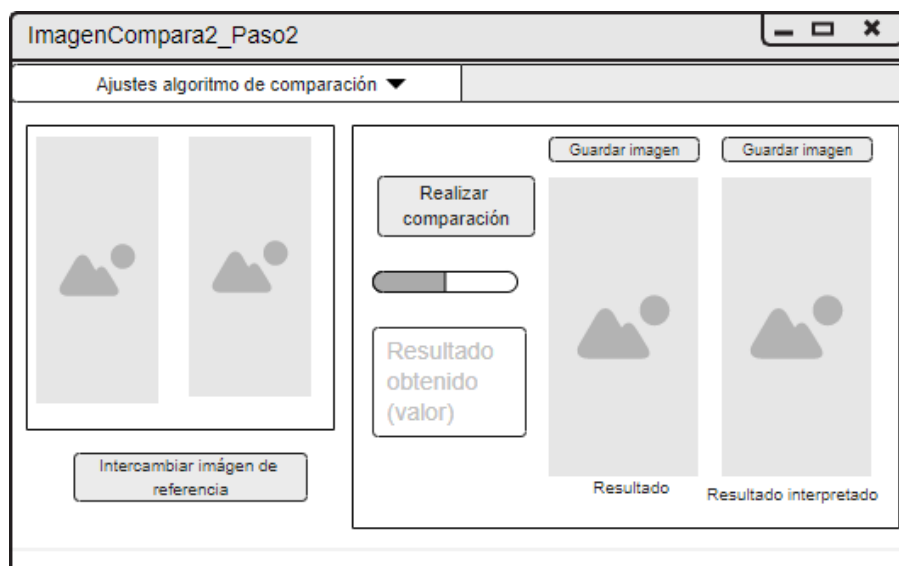


Figura 4.3: Prototipo ventana para realizar la comparación

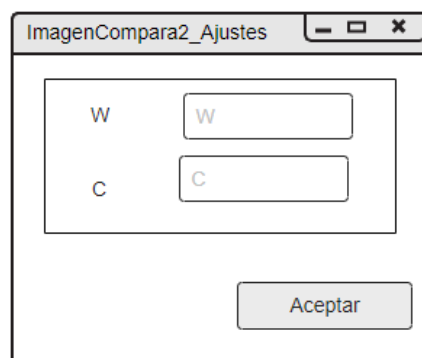


Figura 4.4: Prototipo ventana de ajustes del algoritmo de comparación

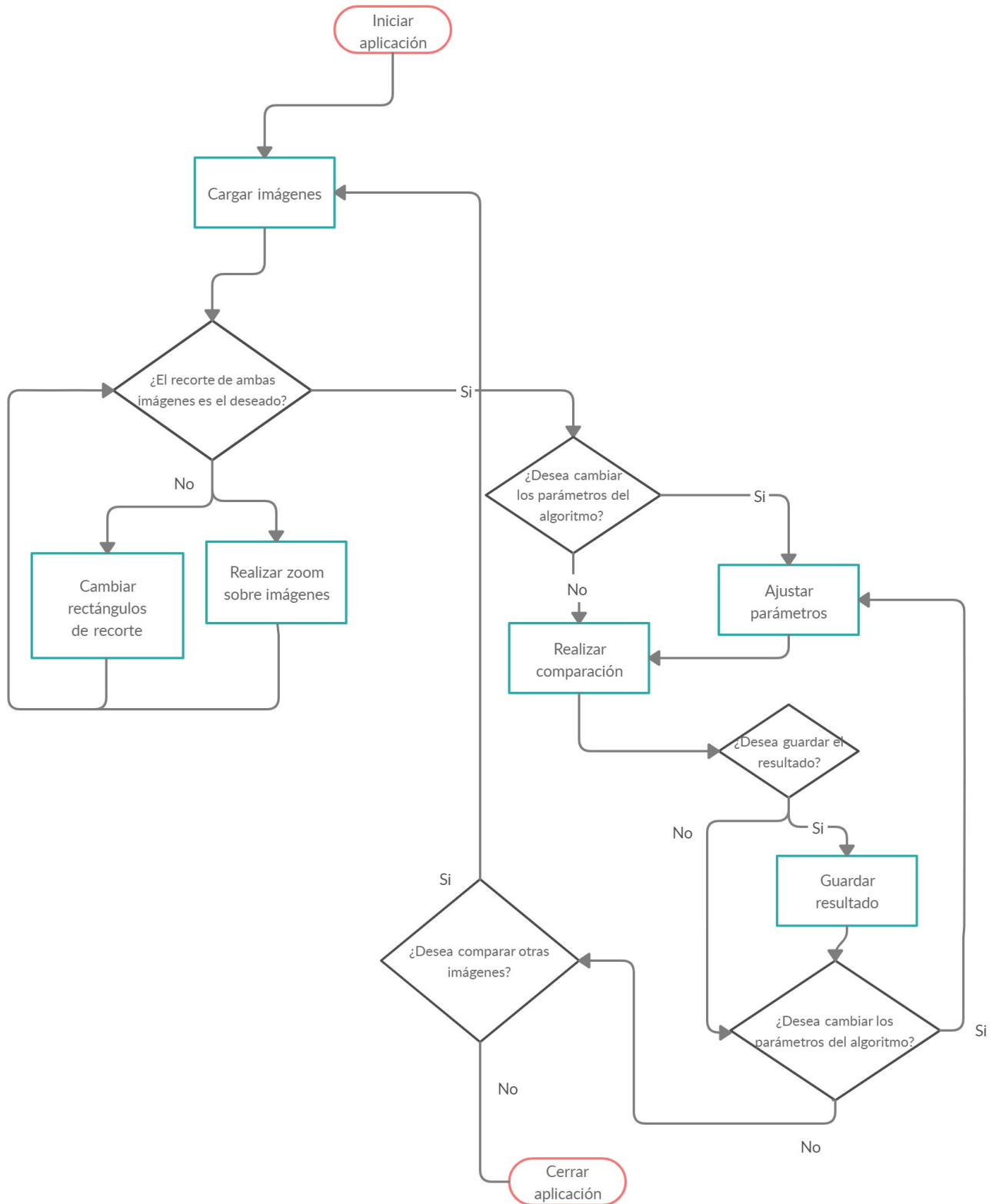


Figura 4.5: Diagrama de flujo de la aplicación

4.2 Tecnología utilizada

Para realizar la comparación de imágenes, no se realiza una alineación lineal de ambas imágenes sino que se ha utilizado el algoritmo de evaluación BIDM [1]. Este algoritmo utiliza el modelo IDM presentado en [2]. El funcionamiento de dicho algoritmo viene detallado en la figura 4.6 obtenida de [1]. Como se puede observar la entradas para ejecutar el algoritmo son la imagen a comparar A ($I \times J$), la imagen de referencia B ($X \times Y$), y los parámetros del modelo IDM que corresponden a w y c para determinar el sesgo de alineamiento y el contexto local de un píxel, respectivamente.

Las derivadas verticales y horizontales utilizadas para la obtención de la diferencia mínima entre dos píxeles son calculadas siguiendo el método detallado en [4]. Para ello, se divide la imagen en una cuadrícula de celdas cuadradas, las cuales contienen la información relativa sobre su nivel de gris normalizado, el nivel de gris de su derivada horizontal y el nivel de gris de su derivada vertical. Para obtener dichos valores suavizados se extiende una ventana de $A \times A$ celdas centrada en la celda actual ponderada por una función gaussiana y las derivadas se calculan por mínimos cuadrados que se ajustan a una función lineal, tal y como se puede observar en la figura 4.6.

El coste temporal del algoritmo es $O(IJw^2c^2)$, dado que se deben recorrer todos los píxeles de la imagen a comparar alineando cada píxel con el píxel de la imagen de referencia cuya diferencia sea mínima con respecto a un píxel que se encuentre a un máximo de w píxeles de su alineamiento lineal, teniendo en cuenta los píxeles de su contexto establecido por c , tal y como se muestra en las figuras 4.7 y 4.8.

```

Input: test image  $A$  ( $I \times J$ ), warp range  $w$ 
reference image  $B$  ( $X \times Y$ ), context window size  $c$ 
Output: BIDM( $w, c$ ) from  $A$  to  $B$ 

 $A^v = \text{ver\_der}(A)$ ;  $A^h = \text{hor\_der}(A)$ ;  $B^v = \text{ver\_der}(B)$ ;  $B^h = \text{hor\_der}(B)$ 
for  $i = 1$  to  $I$  do
  for  $j = 1$  to  $J$  do {
     $i' = \lfloor i \frac{X}{J} \rfloor$ ,  $j' = \lfloor j \frac{Y}{J} \rfloor$ ,  $z = \lfloor \frac{c}{2} \rfloor$ 
     $S_1 = \{1, \dots, X\} \cap \{i' - w, \dots, i' + w\}$ 
     $S_2 = \{1, \dots, Y\} \cap \{j' - w, \dots, j' + w\}$ 

     $\text{map}(i, j) = \min_{\substack{x \in S_1 \\ y \in S_2}} \sum_{m=-z}^z \sum_{n=-z}^z (A_{i+n, j+m}^v - B_{x+n, y+m}^v)^2 + (A_{i+n, j+m}^h - B_{x+n, y+m}^h)^2$ 
  }
  normalize_depth(map, 255)
  binarize(map) //Otsu's method

  fg =  $\{(x, y) \mid A(x, y) < 255\}$  //Foreground pixels
  cp = fg  $\cap \{(x, y) \mid \text{map}(x, y) = 0\}$  //Correct pixels

  return  $\frac{|cp|}{|fg|}$  //Correct pixels ratio

```

Figura 4.6: Algoritmo de evaluación Binary IDM (BIDM)

El algoritmo comienza recorriendo todos los píxeles de la imagen a comparar para realizar el alineamiento desde la esquina superior izquierda estableciendo su ventana de contexto asociada, correspondiente con el recuadro verde en las figuras. La figura 4.7

representa un estado de dicho recorrido, en el cual se quiere alinear un píxel situado sobre el hombro izquierdo con un píxel de la imagen de referencia. Para realizar dicho alineamiento, se calculan los rectángulos de color azul y negro asociados al parámetro W para establecer el mejor alineamiento dentro del sesgo establecido tal y como se muestra en la figura 4.8.

Para ello, la ventana de contexto asociada a la imagen de referencia realiza un recorrido por el rectángulo establecido de color azul para determinar el píxel contenido en dicha ventana (píxel rojo) que obtiene una diferencia menor con respecto al píxel de la imagen a comparar de la figura 4.7 con una distancia máxima de W píxeles entre el píxel a alinear en la imagen de referencia y su correspondiente píxel asociado a un alineamiento lineal (píxeles rojo y morado, respectivamente de la figura 4.8).

Tras obtener la diferencia mínima de cada píxel de la imagen a comparar con respecto a la imagen de referencia, el resultado se normaliza con un rango de 0 a 255, se convierte a binario utilizando el método de Otsu [3] y se realiza la intersección de dicho resultado junto con los píxeles de la imagen a comparar (sin los píxeles correspondientes al fondo) para obtener aquellos que han sido correctamente alineados. Y por consiguiente, los píxeles que han sido alineados incorrectamente.

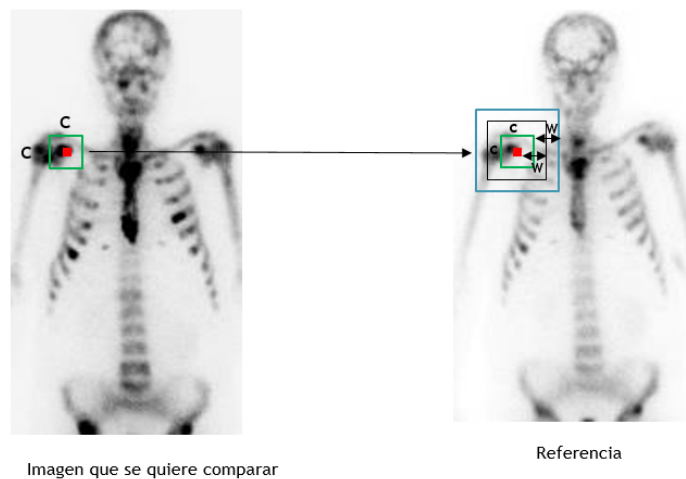


Figura 4.7: Parámetros del algoritmo de evaluación

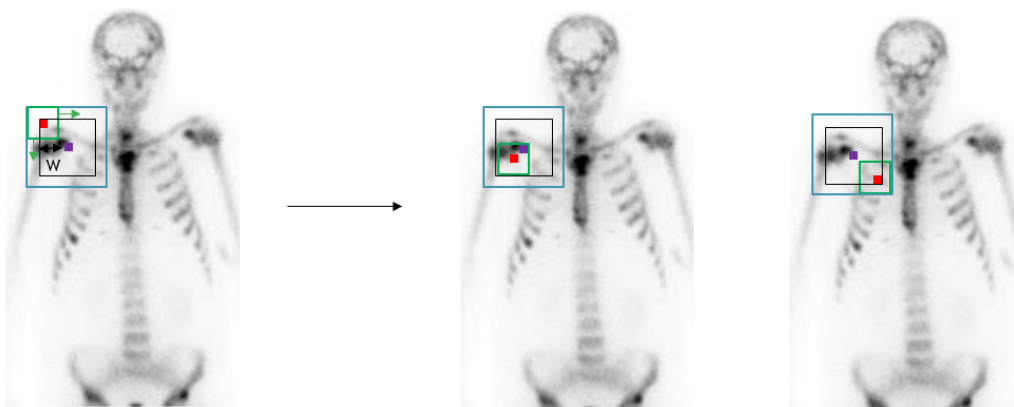


Figura 4.8: Parámetros del algoritmo de evaluación

Tras la ejecución del algoritmo BIDM entre dos imágenes se obtiene el porcentaje asociado a la similitud entre ambas correspondiente al número de píxeles alineados correctamente y una imagen resultado que muestra los píxeles que difieren entre ambas.

En nuestra aplicación, se ha utilizado este algoritmo para realizar la comparación entre las dos imágenes médicas recortadas. Una vez obtenido el resultado correspondiente al valor de similitud y la imagen que muestra los píxeles de variación entre ambas, se ha utilizado este algoritmo para hacer una segunda comparación entre la imagen a comparar y el resultado obtenido de la primera comparación para realizar la interpretación del resultado. Esta segunda comparación se ha llevado a cabo para poder facilitar al personal sanitario a identificar las áreas de la imagen que habían presentado una mejoría o empeoramiento, dado que con la imagen resultado obtenida en la primera comparación, no se podía identificar qué píxeles correspondían a una evolución favorable o desfavorable. En el capítulo siguiente, en el apartado 5.2.4 correspondiente a la clase Comparación se ilustra cómo se generan e interpretan los resultados obtenidos en ambas ejecuciones del algoritmo, dado que dependiendo del tipo de estudio realizado sobre las imágenes a comparar el procedimiento para realizar la interpretación del resultado varía.

Por último, para llevar a cabo la programación de nuestra aplicación, se ha utilizado el entorno de trabajo .NET de Windows para utilizar las clases relativas a la interfaz gráfica y el resto de funcionalidades incorporadas en Visual Studio.

CAPÍTULO 5

Desarrollo de la solución propuesta

En este capítulo, se procede a desarrollar la solución propuesta para nuestra aplicación siguiendo el diseño realizado según los capítulos anteriores.

Para realizar la implementación, se ha llevado a cabo una programación modular dando que los diferentes funcionalidades implementadas en esta interfaz se han asociado en distintos módulos lógicos que interactúan entre ellos a través de las clases implementadas. En la figura 5.1 se puede observar los elementos utilizados en el proyecto. Para desarrollar las ventanas correspondientes a los prototipos diseñados en el capítulo anterior, se han utilizado formularios de Windows (*Windows Forms*). Asimismo, para implementar la funcionalidad, se ha incorporado una clase auxiliar de utilidad correspondiente al archivo de la clase *Imagen* para realizar la edición de las imágenes, y los ejecutables compilados correspondientes al algoritmo de evaluación BIDM utilizado y a la versión portable del comando *convert* de *ImageMagick*. A continuación, en las sucesivas secciones se detalla el desarrollo realizado en profundidad.

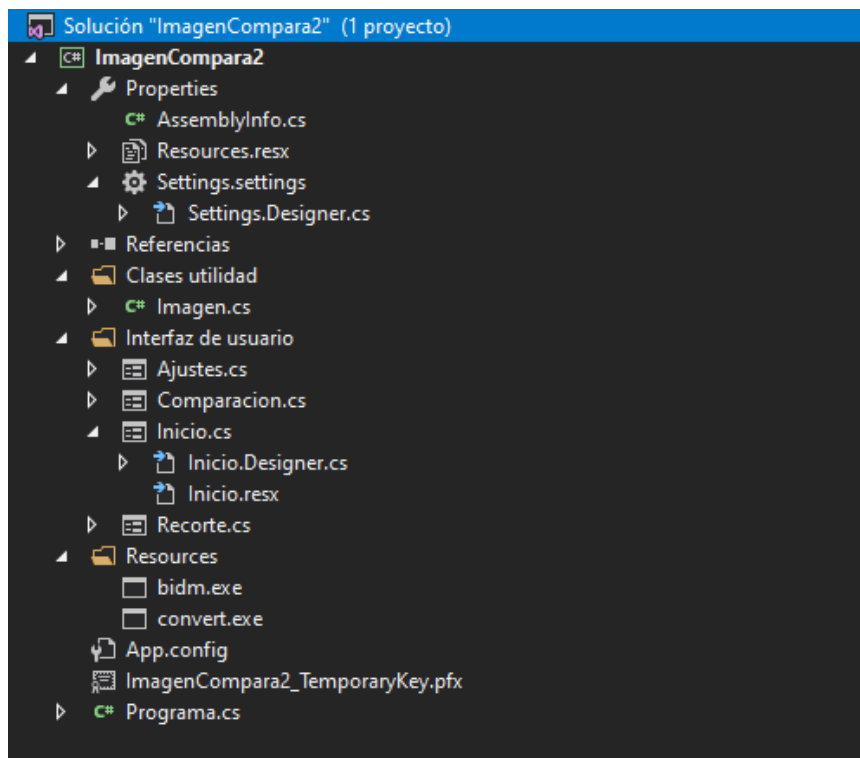


Figura 5.1: Solución ImagenCompara2

5.1 Desarrollo de la interfaz

El desarrollo de la interfaz ha sido realizado siguiendo los prototipos diseñados en el capítulo anterior. Para ello, como ya ha sido mencionado anteriormente, se han utilizado formularios de Windows (*Windows Forms*). Gracias a esta interfaz de .NET, tras la incorporación de nuevos elementos, tales como botones o cuadros de texto, entre otros, se genera código automáticamente correspondiente a la posición del elemento dentro del formulario, además de otras propiedades que facilitan su programación.

A la hora de desarrollar las funcionalidades, desarrollo que se encuentra explicado en profundidad en la próxima sección, se han añadido o cambiado algunos elementos no presentes con respecto al diseño realizado anteriormente para mejorar el uso de esta herramienta. Pero debido a que son pequeños cambios, no ha sido necesario volver a rediseñar la aplicación.

En las sucesivas figuras 5.2, 5.3, 5.4, 5.5, se puede observar las ventanas implementadas correspondientes a la interfaz de nuestra aplicación. En comparación con los prototipos, las ventanas correspondientes al inicio de la aplicación y al ajuste de los parámetros del algoritmo de las figuras 5.2 y 5.5, respectivamente, no se observa ninguna diferencia significativa. En cambio, en la ventana de la figura 5.3, se observa que se han añadido los botones para ajustar la imagen al ancho o al alto del recuadro de visualización y la opción de establecer un punto de referencia en ambas imágenes.

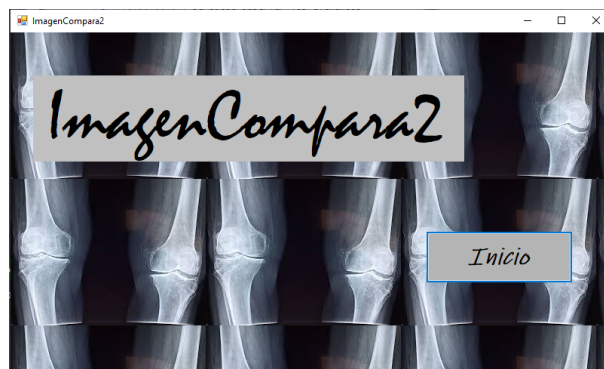


Figura 5.2: Ventana inicio de la aplicación



Figura 5.3: Ventana correspondiente a recortar imagen de la aplicación

Además, en la figura 5.4 se observa que se ha añadido la opción de seleccionar el tipo de comparación que se quiere realizar, según el tipo de imágenes a comparar y se ha añadido la funcionalidad de poder realizar *zoom* sobre las imágenes correspondientes al resultado para facilitar a los médicos la visualización del resultado.

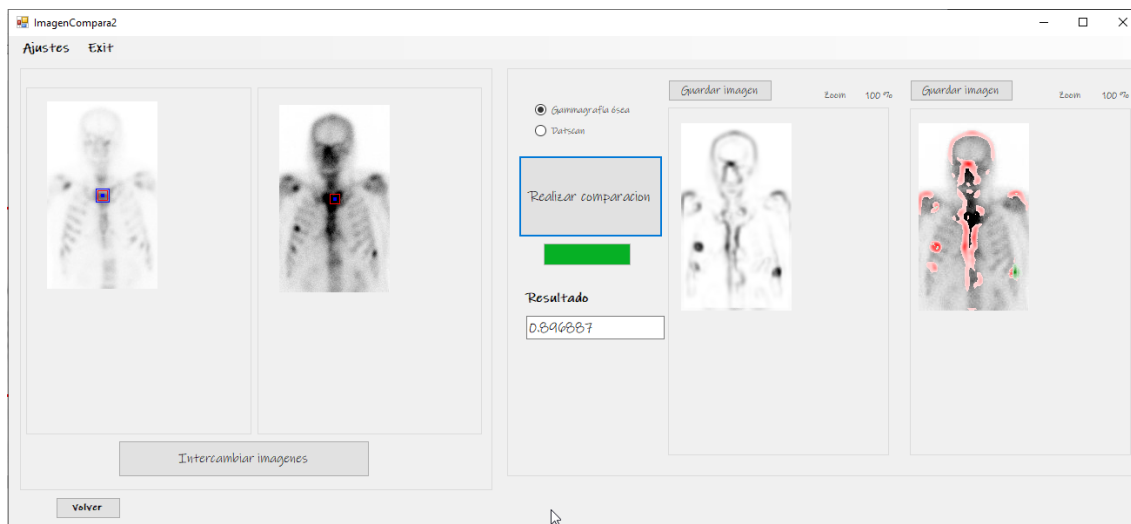


Figura 5.4: Ventana correspondiente a realizar la comparación de la aplicación

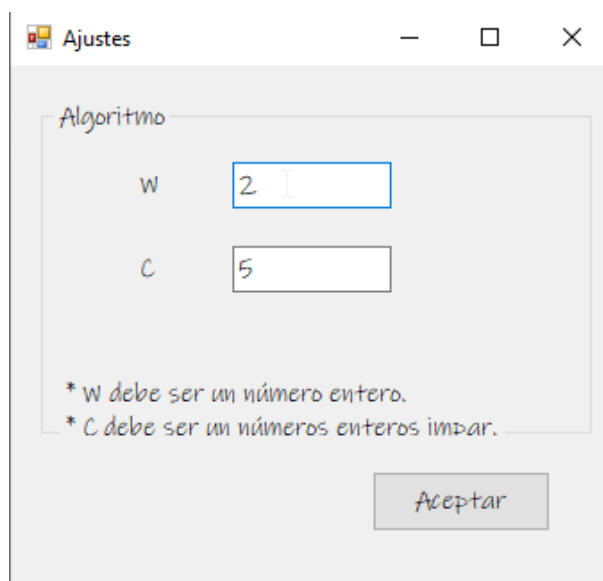


Figura 5.5: Ventana para ajustar los parámetros del algoritmo de la aplicación

Las imágenes utilizadas para cambiar el estilo de la aplicación, es decir la imagen que se encuentra en el fondo de la ventana de inicio o las imágenes correspondientes a las flechas de los botones para mover el rectángulo de recorte, han sido obtenidas de repositorios de imágenes libres de derechos de autor y autorizadas para ser reutilizadas.

5.2 Desarrollo de la funcionalidad

Como se ha detallado en la introducción a este capítulo, para la implementación de la solución de este trabajo se han utilizado los archivos presentes en la figura 5.1. A continuación, se detalla cada clase junto con sus funcionalidades implementadas.

5.2.1. Clase Programa

La clase Programa es la clase principal donde se encuentra el método principal que inicia la aplicación invocando al formulario correspondiente de la ventana de inicio, es decir a la clase Inicio. A continuación se encuentra el código presente en dicha clase.

```
static class Programa
{
    /// <summary>
    /// Punto de entrada principal para la aplicación.
    /// </summary>
    [STAThread]
    static void Main()
    {
        Application.EnableVisualStyles();
        Application.SetCompatibleTextRenderingDefault(false);
        Application.Run(new Inicio());
    }
}
```

Figura 5.6: Clase Programa

5.2.2. Clase Inicio

La clase Inicio presenta la interfaz, y el código asociado, de la ventana de inicio presente en la figura 5.2. Tras clicar sobre el botón de inicio, en el método del evento asociado generado, se guardan los ejecutables compilados en arquitectura de 32 bits del algoritmo de comparación BIDM y el comando Convert de ImageMagick en el directorio actual, para ser ejecutados después para llevar a cabo la comparación y las conversiones de imágenes necesarias. Dado que se deben ejecutar con el uso del terminal, deben encontrarse en el equipo donde se está ejecutando nuestra aplicación. Por tanto, tras iniciar la aplicación se guardan en el directorio actual de trabajo para ser accesibles para su ejecución desde cualquier equipo. De esta forma, no es necesario que se realice ninguna instalación previa de algún programa para poder ejecutar nuestra aplicación, puesto que dicha aplicación ya incorpora todos los elementos necesarios para poder ejecutarse en cualquier equipo que tenga una arquitectura de 32 bits o superior. El código asociado se puede observar en la figura 5.7.

Tras guardar ambos ejecutables, se muestra la ventana correspondiente a la figura 5.3 asociada a la clase Recorte, en la cual se le encarga de cargar y seleccionar el área de interés de cada imagen a comparar.

```
private void buttonInicio_Click(object sender, EventArgs e)
{
    byte[] bidmexe = Properties.Resources.bidm;
    byte[] convertexe = Properties.Resources.convert;
    System.IO.File.WriteAllBytes("convert.exe", convertexe);
    System.IO.File.WriteAllBytes("bidm.exe", bidmexe);

    Form uno = new RecortarImagen();
    uno.Show();
    this.Visible = false;
}
}
```

Figura 5.7: Clase Inicio - Guardar ejecutables necesarios para la aplicación

5.2.3. Clase Recorte

La clase Recorte corresponde a la implementación de la interfaz de ventana de la figura 5.3. En dicha clase se encuentran implementados los requisitos funcionales de cargar imagen, recortar imagen y realizar zoom a la imagen para poder seleccionar las imágenes que el usuario quiere comparar y establecer el área de comparación. A continuación, se encuentra detallada la implementación de cada requisito implementado en esta clase.

Cargar imagen

Para abrir y cargar la imagen en el recuadro de visualización correspondiente en la interfaz, tras pulsar el botón de cargar imagen situado en la parte superior de cada recuadro de visualización en la figura 5.3, a través del su evento de clic asociado, se abre un cuadro de diálogo correspondiente a la clase OpenFileDialog[14]. A través de dicho cuadro de diálogo se puede abrir la imagen deseada cuyo formato sea compatible con la clase ImageFormat [8]. Los formatos compatibles son BMP, EMF, EXIF, GIF, GUID, ICON, GUID, ICON, JPEG, MemoryBmp, PNG, TIFF Y WMF. Aunque para el uso de nuestra aplicación utilizamos principalmente imágenes en formato JPEG y PNG.

Una vez ha sido abierta la imagen con éxito, es asignada al cuadro de visualización correspondiente y se muestra un rectángulo de recorte por defecto con una anchura y altura de 50 y 100 píxeles respectivamente. En caso de comparar imágenes con menor número de píxeles en alguna dimensión, el rectángulo se reajusta a la dimensión máxima correspondiente.

Recortar imagen

Para recortar la imagen a comparar, a través de los elementos de la interfaz, se puede modificar la posición, la anchura y la altura del rectángulo de recorte. Para modificar la posición del rectángulo de recorte, una opción implementada es pulsar los botones asociados a las cuatro acciones que se pueden realizar, las cuales son subir, bajar, mover hacia la derecha y mover hacia la izquierda, representadas con flechas. El evento asociado a cada botón, realiza un desplazamiento del rectángulo de recorte de cinco píxeles en la dirección asociada a cada botón. Otra opción corresponde a pulsar con el ratón sobre un píxel de la imagen. El evento asociado sitúa el vértice superior izquierdo del rectángulo de recorte en el píxel seleccionado manteniendo sus dimensiones. En caso de que el rectángulo de recorte sobrepase las dimensiones de la imagen, éste cambia sus dimensiones para ajustarse a las dimensiones máximas de la imagen.

Por otra parte, para modificar la anchura y la altura, una opción implementada es pulsar los botones de «+» y «-» asociados a cada parámetro. Tras pulsar dichos botones, se aumenta o disminuye cada valor cinco píxeles. También, se puede modificar la anchura o altura cambiando el valor del cuadro de texto asociado si se quiere realizar un cambio más ajustado. Dicho cuadro de texto tiene asociado un evento que detecta cuando se realiza un cambio sobre él y asigna el valor que contiene. En caso de que la cadena de texto contenida en dicho cuadro de texto sea la cadena vacía, el valor asociado al parámetro correspondiente será cero.

Además, otra implementación para ajustar los parámetros de anchura y altura corresponde a pulsar sobre un píxel de la imagen, arrastrar sobre cualquier dirección y soltar la tecla del ratón en otro píxel distinto. El evento asociado a dichas acciones, establece el rectángulo de recorte resultante cuyos vértices opuestos corresponden al píxel seleccionado al pulsar sobre la imagen y al píxel seleccionado con el cursor tras soltar la tecla del ratón tal y como se puede observar en el caso de la figura 5.8. Para ello, en primer lugar se establece el vértice superior izquierdo como el píxel seleccionado al pulsar y se compara las coordenadas del píxel donde se encuentra el ratón al soltar para establecer el vértice superior izquierdo del rectángulo con el valor mínimo de cada dimensión.

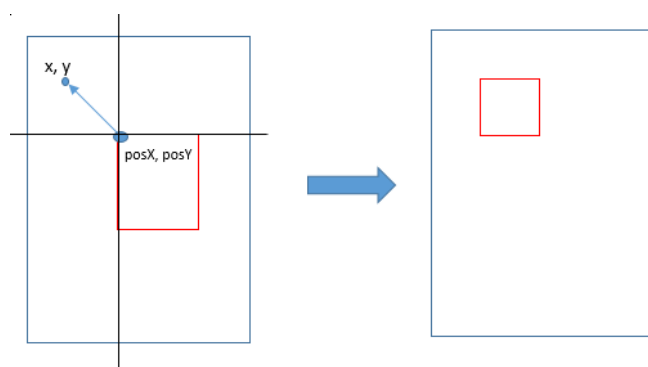


Figura 5.8: CambiarRectángulo

Además, en el caso de que se quieran comparar dos imágenes con las mismas dimensiones, es decir, los rectángulos de recorte tengan las mismas dimensiones, otra funcionalidad implementada para ayudar a ajustar su posición en ambas imágenes es el establecimiento de un punto de referencia en una imagen con respecto a su rectángulo de recorte tal y como se muestra en las figuras 5.9 y 5.10. Para ello, en primer lugar, el usuario debe marcar la opción de establecer un punto de referencia asociado a la imagen. A continuación, tras pulsar sobre un píxel de la imagen, éste es guardado como referencia y se dibuja con un lápiz rojo de la clase Pen [15]. El punto de referencia puede establecerse sobre cualquier píxel de la imagen que se encuentre fuera o dentro del área a recortar. Después, para ajustar el rectángulo de recorte de la otra imagen, se debe clicar sobre el punto sobre el que se quiere establecer su nuevo rectángulo de recorte asociando dicho punto al punto de referencia de la otra imagen y guardando la relación de la distancia establecida entre dicho punto y su rectángulo de recorte. Para ello, tras seleccionar dicho punto, se calcula la distancia relativa entre el punto establecido como referencia y su rectángulo de recorte, y en la segunda imagen a través del píxel seleccionado, teniendo en cuenta dicha distancia relativa, se establece el nuevo rectángulo de recorte.

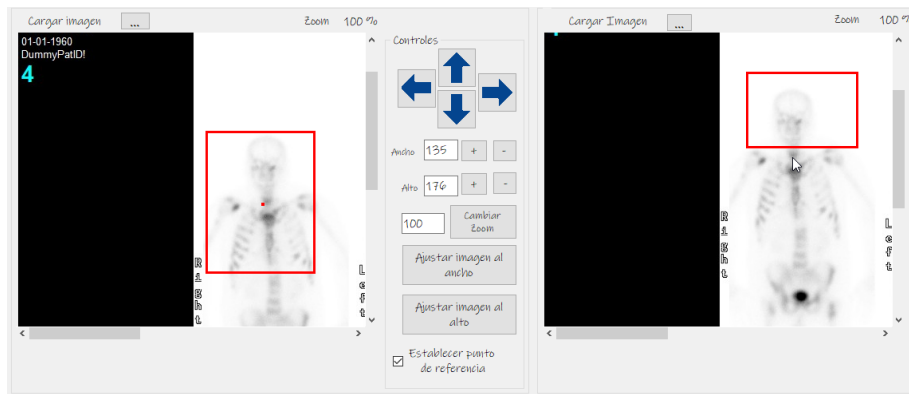


Figura 5.9: Establecer punto de referencia - Paso 1

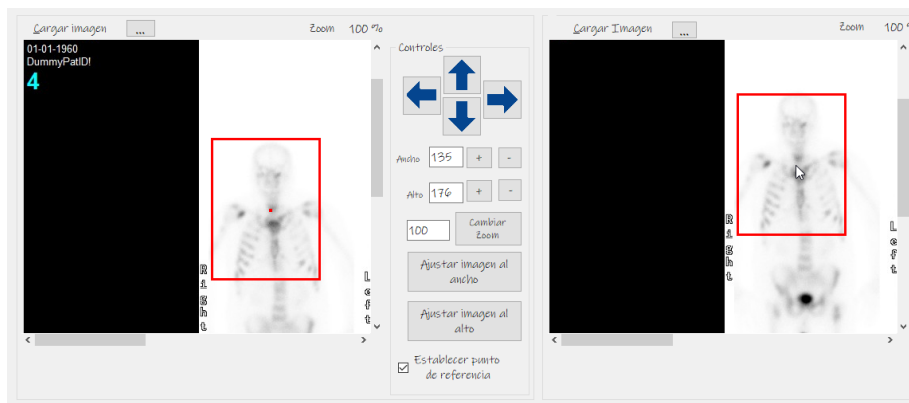


Figura 5.10: Establecer punto de referencia - Paso 2

Realizar *zoom* a la imagen

Para realizar *zoom* a la imagen, una opción implementada es tras situar el cursor del ratón sobre el cuadro de visualización de la imagen, mover la rueda del ratón en ambos sentidos para aumentar o disminuir la imagen. El evento captura dicho movimiento y realiza el *zoom* correspondiente. Además, otra opción es a través de cambiar el valor asociado al cuadro de texto asociado al *zoom*. Tras modificar su valor, se debe pulsar el botón de cambiar *zoom* asociado y las dimensiones de la imagen cambian.

Dado que tras la realización sucesiva de *zoom* sobre una imagen a través del método de la figura 5.11, se pierde información al hacer la interpolación, al cargar la imagen, nos guardamos la imagen original y se realiza el *zoom* sobre ella y no sobre la imagen ya modificada. Además, como se puede observar en dicho método, si tras realizar una disminución de la imagen, la anchura de la imagen es inferior a 70 píxeles, no se realiza ninguna acción sobre ella para evitar imágenes con dimensiones muy pequeñas que dificulten su visualización a través de la interfaz.

```
public Image ZoomImage(Image img, Size size)
{
    int imgOX = img.Width;
    int imgOY = img.Height;
    if (img.Width + (img.Width * size.Width / 100) >= 70)
    {
        Bitmap bm = new Bitmap(img, img.Width + (img.Width * size.Width / 100), img.Height + (img.Height * size.Height / 100));
        Graphics gpu = Graphics.FromImage(bm);
        gpu.InterpolationMode = System.Drawing.Drawing2D.InterpolationMode.HighQualityBicubic;
        anchMax = bm.Width;
        altMax = bm.Height;

        return bm;
    }
    return null;
}
```

Figura 5.11: Método *zoom*

Además, al cambiar las dimensiones de la imagen, el rectángulo de recorte asociado también cambia sus dimensiones para que la información del área recortada no varíe, aunque sí sus dimensiones. Por tanto, tras realizar *zoom* sobre una imagen, de forma proporcional, se modifican las dimensiones y la posición del vértice superior izquierdo del rectángulo de recorte de forma proporcional al *zoom* realizado.

Una vez el usuario ha ajustado las imágenes que desea recortar editando el rectángulo de recorte y las dimensiones de cada imagen, tras pulsar el botón de siguiente, el evento asociado comprueba que se han cargado ambas imágenes y presentan un rectángulo de recorte válido. En dicho caso, se abre la ventana correspondiente para hacer la comparación correspondiente con la clase Comparación. En caso contrario, la ventana correspondiente para hacer la comparación, no será mostrada.

5.2.4. Clase Comparación

La clase Comparación, correspondiente a la ventana para realizar la comparación y mostrar los resultados obtenidos asociada a la figura 5.4. Además, es posible guardar las imágenes pertenecientes al resultado del algoritmo y a su interpretación generadas tras realizar la comparación para que el usuario, en este caso, el personal sanitario, pueda guardarse dicha información en su equipo para futuras consultas. Por tanto, en esta clase se encuentran implementados los requisitos funcionales correspondientes a realizar comparación y guardar imagen.

Realizar comparación

Para realizar la comparación entre las imágenes recortadas en la ventana anterior, se debe ejecutar el algoritmo de evaluación BIDM explicado en la sección 4.2 pasándole como parámetros la imagen a comparar, la imagen referente, los parámetros correspondientes a la ventana de contexto y de distorsión, además de el parámetro opcional de mostrar el resultado obtenido en una imagen cuyo nombre es especificado en dicho parámetro.

Para ello, tras pulsar el botón asociado a realizar la comparación, se guardan las imágenes recortadas en el directorio actual de trabajo de la aplicación, siendo la imagen referente guardada como imagen antes (imagen situada más hacia izquierda) y la imagen a comparar como imagen después (segunda imagen) en formato JPEG. El algoritmo debe ser llamado con imágenes en formato PGM [16] en el formato P2. Las imágenes en formato PGM pueden estar codificadas en dos tipos, correspondientes a P2 y P5 [5]. El formato P2 codifica el valor de los píxeles en código ASCII y el formato P5, en binario. La implementación del algoritmo BIDM utilizado no es de nuestra autoría y dado que dicha implementación fue realizada inicialmente para el sistema operativo Linux, tras realizar su invocación con imágenes en formato P5, la codificación de los caracteres especiales en

Windows varía con respecto a Linux y por tanto, se produce un error. Para evitarlo, se realiza la llamada del algoritmo con imágenes en formato PGM en el formato P2.

El formato habitual es P5, por tanto, tras convertir la imagen en formato JPEG a PGM utilizando el ejecutable correspondiente al comando de *Convert* de ImageMagick, se debe realizar una segunda conversión para cambiar su formato. Para ello, se invoca el ejecutable *Convert* guardado con anterioridad en el directorio actual y se le pasan los argumentos necesarios a través de la creación de un proceso de la clase *Process* [17]. En la figura 5.12 se observa el fragmento de código correspondiente para la invocación del ejecutable *Convert* para cambiar el formato a través de la opción «*--compress None*».

```
var p3 = new System.Diagnostics.Process();
p3.StartInfo.UseShellExecute = false;
p3.StartInfo.RedirectStandardOutput = true;
p3.StartInfo.CreateNoWindow = true;
p3.StartInfo.FileName = "convert.exe";
p3.StartInfo.Arguments = "imagenDespues" + contador + ".pgm -compress None imagenDespues_" + contador + ".pgm";

p3.Start();
p3.WaitForExit();
```

Figura 5.12: Fragmento de código correspondiente a la conversión de la imagen de formato P5 a P2

Una vez ambas imágenes han sido convertidas al formato P2, se realiza la invocación del ejecutable *BIDM* correspondiente al algoritmo de comparación. En la figura 5.13 se observa el fragmento de código utilizado para su invocación, donde se pueden observar que la imagen resultante de la comparación realizada entre la imagen a comparar y la imagen de referencia con el valor correspondiente de los parámetros *W* y *C*. La ejecución del algoritmo no es biyectiva, es decir, dadas dos imágenes *A* y *B*, no se obtiene el mismo resultado al comparar la imagen *A* contra la imagen *B* y viceversa. Por tanto, el usuario debe elegir qué imagen quiere tomar como imagen a comparar y como imagen de referencia. En la interfaz la imagen situada más a la izquierda corresponde a la imagen de referencia y la imagen situada a la derecha de ésta, es la imagen a comparar. En caso de que se quieran cambiar ambas imágenes, el evento asociado al botón situado en la parte inferior de ambas imágenes realiza dicho cambio.

```
var p = new System.Diagnostics.Process();
p.StartInfo.UseShellExecute = false;
p.StartInfo.RedirectStandardOutput = true;
p.StartInfo.CreateNoWindow = true;
p.StartInfo.FileName = "bidm.exe";
p.StartInfo.Arguments = "imagenDespues_" + contador + ".pgm imagenAntes_" + contador + ".pgm " + w + " " + c + " -m res.pgm";
p.Start();

string output = p.StandardOutput.ReadToEnd();
p.WaitForExit();
progressBar.Increment(5);
```

Figura 5.13: Fragmento de código correspondiente a la llamada del algoritmo de comparación

Dado el caso de que no se pueda realizar la comparación entre ambas imágenes, se mostrará un error en el cuadro de texto asociado al valor del resultado y en caso contrario, se mostrará el resultado obtenido de la semejanza de ambas imágenes en tanto por uno tras la ejecución del algoritmo. Además, si la comparación se ha realizado con éxito, en los cuadros de visualización situados en la mitad derecha aparecen en primer lugar la imagen resultado obtenida tras la comparación, en la cual se muestra con píxeles de color negro aquellos que han sido variados con respecto a la imagen referencia y en segundo lugar, la imagen correspondiente a la interpretación del resultado con respecto a la imagen a comparar.

Para realizar la interpretación del resultado, se tiene en cuenta el tipo de imágenes a comparar, es decir, si se trata de imágenes obtenidas tras realizar una gammagrafía ósea o a través de la prueba DaTSCAN, dado que la interpretación es contraria. Para ello, antes de realizar la comparación, el usuario debe marcar la opción según las imágenes a comparar.

Gammagrafía ósea

En el caso de que las imágenes a comparar se traten de imágenes obtenidas tras realizar una gammagrafía ósea como en las figuras 5.14 y 5.15, los píxeles de variación de la imagen resultado, corresponden a las píxeles que han variado en la segunda imagen con respecto a la primera. Como se puede observar entre ambas figuras, si se intercambian ambas imágenes, el resultado obtenido en la tercera imagen difiere.

Para realizar la interpretación de la cuarta imagen se utiliza el procedimiento reflejado en la figura 5.16. El resultado obtenido tras la comparación es solapado a la segunda imagen o imagen a comparar y se realiza una segunda comparación auxiliar entre ambas imágenes para obtener los píxeles que han disminuido su nivel de gris con respecto a la imagen de referencia original. Dicho resultado se corresponde con la imagen asociada al resultado 2. Finalmente, si a la imagen a comparar 1 se le solapa el resultado 1, correspondiente a la primera comparación, en rojo y el resultado 2 en verde, en la imagen interpretada aparecen en rojo los píxeles que han aumentado su nivel de gris y en verde los píxeles que han disminuido dicho nivel con respecto a la imagen de referencia 1. En nuestra aplicación seguimos este procedimiento dado que si en la imagen a comparar el nivel de gris disminuye, esto significa que el paciente ha mejorado y se debe mostrar en verde y en caso contrario, si se ha producido un aumento del nivel de gris, el paciente ha empeorado y dicha área se debe mostrar en rojo.

Aún así, cabe destacar que no se puede afirmar que todos los píxeles mostrados en rojo correspondan con un empeoramiento ni que todos los píxeles verdes correspondan con una mejora, dado que el resultado obtenido varía significativamente según los parámetros establecidos del algoritmo. Para ello, se debe realizar el ajuste correcto para cada par de imágenes a comparar e interpretar dichos píxeles como un posible empeoramiento o mejora.

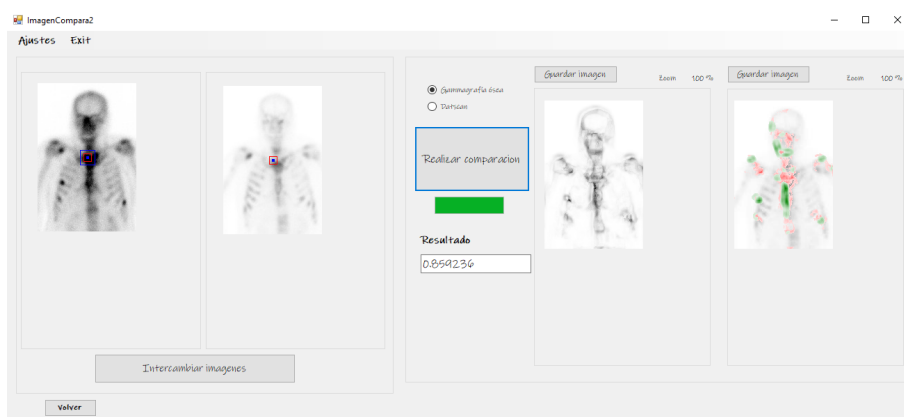


Figura 5.14: Comparación gammagrafía ósea 1

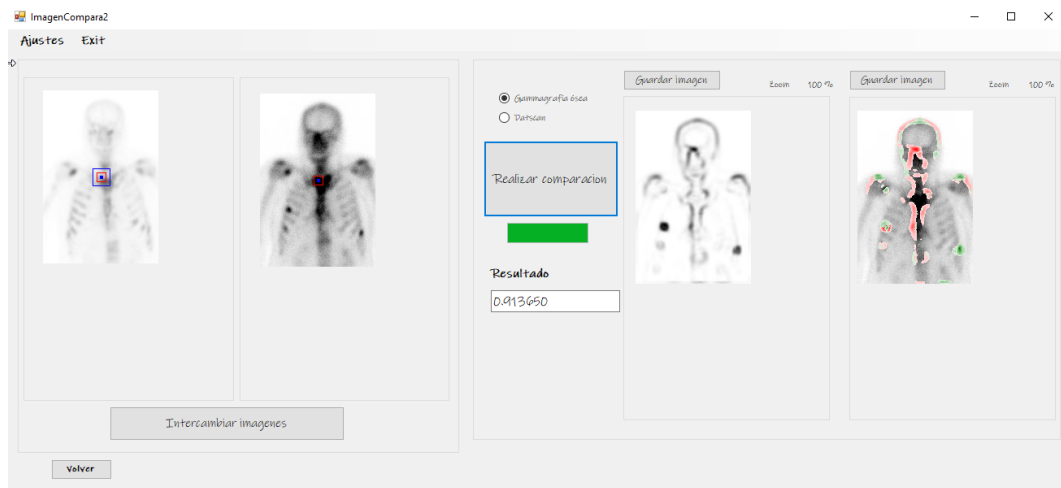


Figura 5.15: Comparación gammagrafía osea 2

PRIMERA COMPARACIÓN

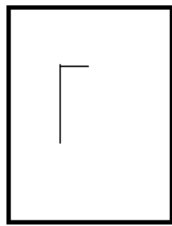


Imagen de referencia 1

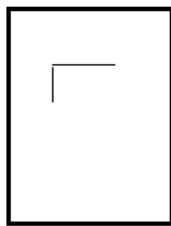
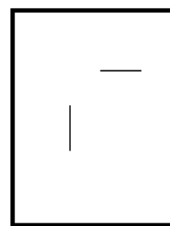


Imagen a comparar 1



Resultado 1

SEGUNDA COMPARACIÓN

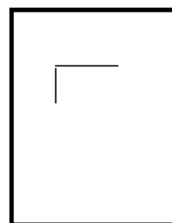


Imagen de referencia 2
(imagen a comparar 1)

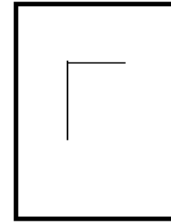
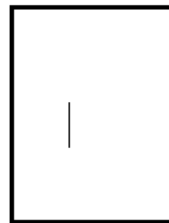


Imagen a comparar 2
(imagen a comparar 1 + Resultado 1)



Resultado 2

OBTENCIÓN IMAGEN INTERPRETADA

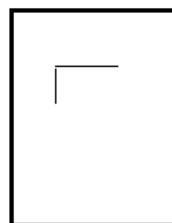


Imagen a comparar 1

+



Resultado 1

+



Resultado 2

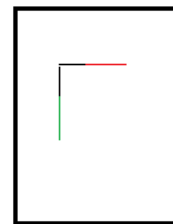


Imagen interpretada

Figura 5.16: Procedimiento para la obtención de la imagen interpretada

DaTSCAN

En el caso de que las imágenes a comparar se traten de imágenes obtenidas tras realizar la prueba DaTSCAN como en las figuras 5.17 y 5.18, ambas imágenes se convierten a escala de grises y el procedimiento implementado para interpretar la imagen resultado se corresponde al procedimiento de la figura 5.16 pero con los colores de los resultados obtenidos invertidos. Esto es debido a que si en la imagen a comparar se disminuye el nivel de gris con respecto a la imagen de referencia, el paciente empeora y en caso contrario, si se aumenta el nivel de gris, el paciente mejora dado que presenta una mayor actividad.

De igual forma que en el caso anterior, los píxeles rojos y verdes no aseguran que exista un empeoramiento o mejora, respectivamente. Por tanto se deben interpretar como un posible empeoramiento o mejora y cambiar los parámetros de ajuste para obtener resultados óptimos según cada par de imágenes a comparar.

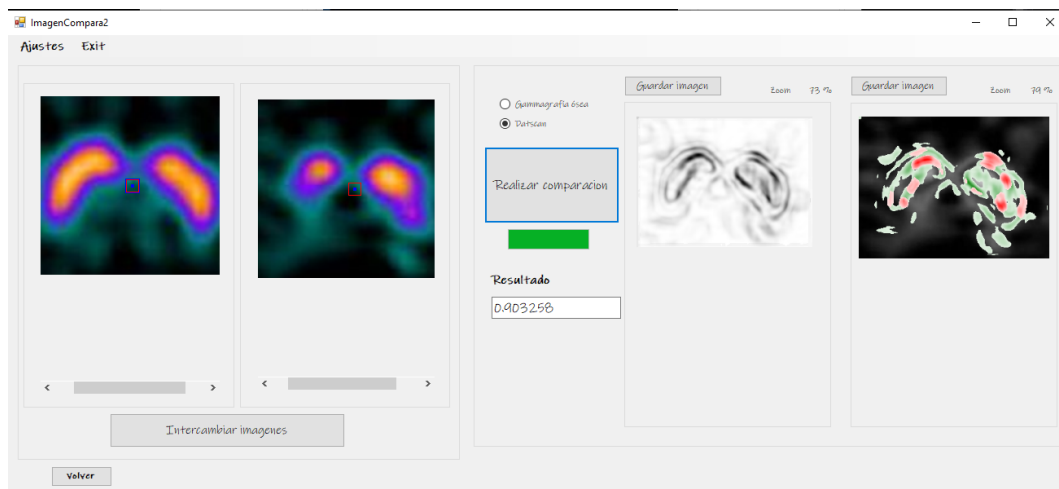


Figura 5.17: Comparación DaTSCAN 1

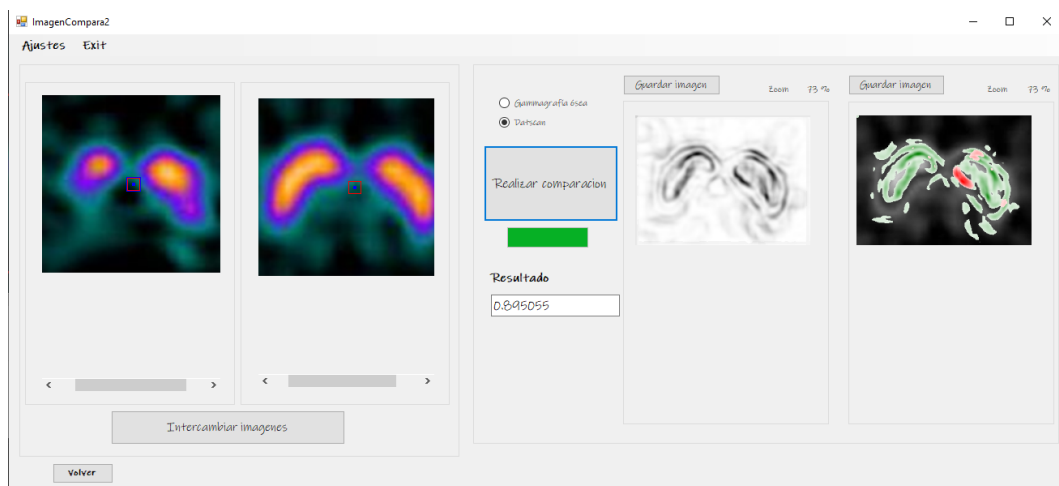


Figura 5.18: Comparación DaTSCAN 2

Guardar imagen

Para guardar las imágenes correspondientes con el resultado obtenido de la comparación y la imagen de la interpretación del resultado, tras pulsar el botón de guardar imagen asociado a cada imagen y situado en su parte superior, se abre un cuadro de diálogo correspondiente a la clase SaveFileDialog [18] donde el usuario puede seleccionar la ruta y establecer el nombre de la imagen a guardar. Por defecto, si no se especifica un formato, la imagen se guarda en formato JPEG pero también se ha implementado que se pueda elegir la opción de guardar la imagen en formato PNG o BMP.

5.2.5. Clase Ajustes

La clase Ajustes se corresponde con la implementación de la interfaz de la ventana de la figura 5.5 correspondiente a la modificación de los parámetros del algoritmo de comparación BIDM para ajustar la comparación realizada. Si se tienen en cuenta los requisitos funcionales, esta clase implementa el requisito de ajustar los parámetros del algoritmo de comparación.

Ajustar parámetros del algoritmo de comparación

Para ajustar los parámetros del algoritmo de comparación, por restricciones del algoritmo el valor asociado a W correspondiente con el área de distorsión debe ser un número entero y el valor asociado a C , correspondiente con la ventana de contexto debe ser un número entero impar. Para ello, tras modificar los valores asociados a cada parámetro en los cuadros de texto asociados, tras pulsar el botón de aceptar, se comprueba si se cumple dicha restricción. En dicho caso, los parámetros son cambiados y la ventana se cierra. Pero en caso contrario, si el valor modificado para el parámetro C corresponde con un número par, se incrementa su valor en una unidad. Asimismo, en caso de que ambos parámetros sean inválidos, se muestra un error y la ventana no se cierra.

5.2.6. Clase Imagen

Por último, la clase Imagen, dentro de la carpeta de clases útiles, es una clase utilidad que representa el objeto de una imagen que es mostrada y puede ser editada a través de la interfaz. Sus atributos y métodos implementados se muestran en la clase de la figura 5.19. Entre sus atributos se encuentra el contenido de la imagen en un mapa de bits y sus dimensiones asociadas. Además, para representar el rectángulo de recorte asociado a cada imagen, las coordenadas del vértice situado en la esquina superior izquierda del rectángulo de recorte junto con sus dimensiones, corresponden a los atributos posX, posY, ancho y alto. Con dicha información, el rectángulo de recorte puede ser construido y mostrado a través de la interfaz. En caso de que se establezca un punto de referencia para ayudar a ajustar el rectángulo de recorte, las coordenadas de dicho punto corresponden con los atributos de refX y refY.

Los métodos implementados corresponden a todas las acciones que se pueden realizar sobre dicha imagen, tales como abrir la imagen, realizar zoom sobre ella, dibujar los elementos gráficos correspondientes a los rectángulos de recorte y píxeles en la imagen, mover y ajustar el rectángulo de recorte, y guardar la imagen en un directorio establecido, asociadas a los eventos explicados en el capítulo anterior.

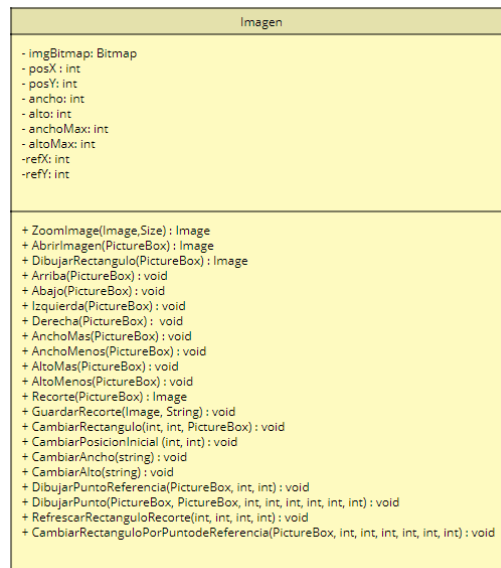


Figura 5.19: Clase Imagen

CAPÍTULO 6

Pruebas

En este capítulo, se van a realizar distintas pruebas sobre el caso de un paciente anónimo (figura 6.1). El objetivo es comprobar cómo cambia el resultado obtenido al variar las dimensiones del rectángulo de recorte o de la propia imagen, realizando *zoom* sobre ella, y al variar el valor de los parámetros W y C del algoritmo.

Además, también se calculará el coste temporal total de la comparación, es decir, se calculará el tiempo transcurrido desde que se pulsa el botón de comenzar la comparación hasta que se muestra el resultado completo tras realizar todas las comparaciones y conversiones de imágenes necesarias. Para realizar dicho cálculo, se ha añadido el código necesario y el tiempo transcurrido es mostrado en segundos a través de la interfaz.

En estudios anteriores donde el algoritmo de evaluación BIDM fue utilizado para reconocer expresiones matemáticas [1], se pudo observar que se obtenían mejores resultados con valores de W y C relativamente bajos con respecto a las dimensiones de la imagen. Además, dada una imagen a comparar cuyas dimensiones sean $I \times J$, la complejidad temporal del algoritmo BIDM es $O(IJw^2c^2)$ [1], por tanto la reducción de los valores de W y C reducirán el coste notablemente. En las pruebas siguientes, se tienen en cuenta estas restricciones para obtener un resultado óptimo.

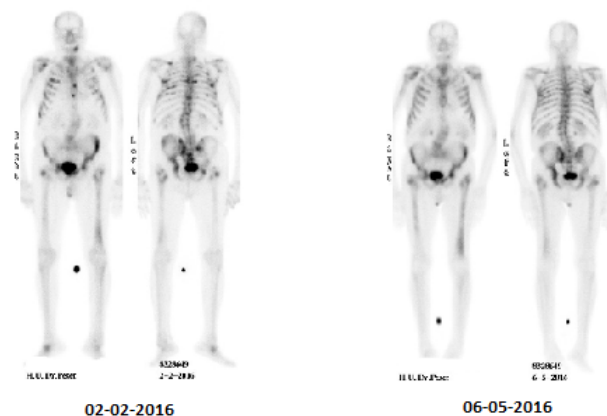


Figura 6.1: Paciente Anónimo

Como se puede observar en la figura 6.1 el paciente presenta una leve mejoría en la imagen correspondiente al 6 de mayo con respecto a la imagen del 2 de febrero, dado que diversos píxeles presentes en la zona de la columna vertical y las costillas han disminuido su nivel de gris, por tanto, en las pruebas sucesivas se espera visualizar dicha mejoría.

6.1 Resultados variando las dimensiones del rectángulo de recorte y de la imagen

En primer lugar, como se puede observar en la figura 6.2, se ha recortado la imagen completa con un rectángulo de recorte de 364 píxeles de ancho y 611 píxeles de alto y se ha realizado la comparación con los parámetros del algoritmo establecidos por defecto ($W=5$ y $C=5$). En dicha figura se puede observar que la comparación ha tardado 6,43 segundos y en la imagen interpretada sólo se observa una variación entre ambas imágenes en la zona pélvica.

En este caso, no se observa la mejoría detectada anteriormente pero el coste temporal es aceptable.

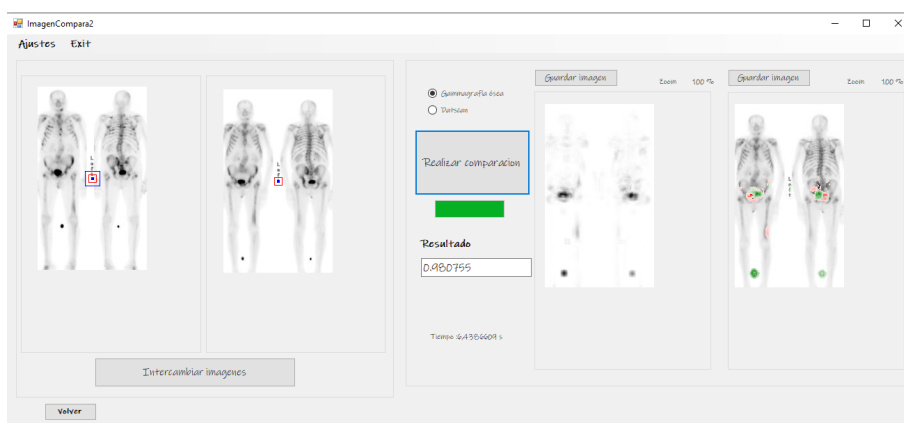


Figura 6.2: Comparación $W=5$ y $C=5$, imagen 100 %

Si realizamos un *zoom* a ambas imágenes de 200 %, las dimensiones del rectángulo de recorte cambian a 729 píxeles de ancho y 1222 píxeles de alto. El resultado obtenido se encuentra en la figura 6.3, en la cual se puede observar que la comparación ha tardado 97,53 segundos y la imagen interpretada sigue presentando sólo una pequeña variación en la zona pélvica, pero menos significativa.

En este caso el tiempo transcurrido para llevar a cabo la comparación y la obtención de la interpretación de los resultados ha aumentado significativamente debido a que las dimensiones de la imagen son relativamente altas. Se podría afirmar que estas dimensiones junto con el valor de los parámetros del algoritmo no obtienen un buen resultado dado que la mejoría no se ve reflejada en la imagen y además, el coste es muy elevado.

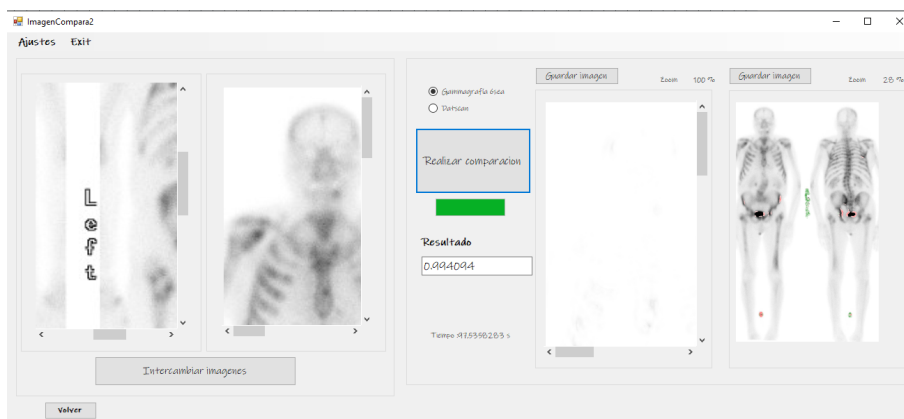


Figura 6.3: Comparación $W=5$ y $C=5$, imagen 200 %

Dado que haciendo un recorte de toda la imagen, no obtenemos los resultados esperados, vamos a centrarnos en la posible mejoría observada a priori, cambiamos el rectángulo de recorte focalizando la zona de las costillas con un rectángulo de recorte de 290 píxeles de ancho y 344 píxeles de alto. En la figura 6.4 correspondiente a dicha comparación se puede observar que el tiempo transcurrido es de 12,93 segundos y en la imagen interpretada si se observa una variación en los píxeles relacionados con la posible mejora pero se muestran en rojo. Sólo aparece de color verde la mejora de la última costilla izquierda.

Cabe recordar que no se han cambiado los parámetros por defecto del algoritmo, los cuales dependiendo de la imagen pueden no ser los óptimos, debido a esto, en el siguiente apartado, se cambian dichos parámetros para observar cómo afecta al resultado.

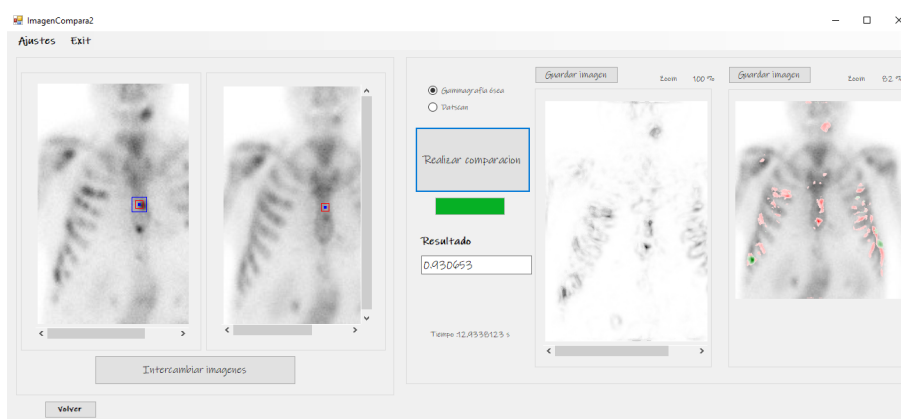


Figura 6.4: Comparación $W=5$ y $C=5$ de zona focalizada, imagen 200%

6.2 Resultados variando los parámetros del algoritmo

Para comprobar cómo afecta la variación de los parámetros del algoritmo en este caso particular, se han realizado cuatro comparaciones correspondientes a valores de W y C relativamente bajos, valores de W y C relativamente altos, valores de W menores que C y valores de W mayores que C . Las imágenes comparadas se corresponden con las imágenes del último de la sección anterior, dado que en dicho caso se habían obtenido mejores resultados que en el resto.

En primer lugar, si se establecen valores para W y C bajos, tales como $W=1$ y $C=1$, se observa el resultado presente en la figura 6.5. En dicha imagen, se puede observar que el tiempo transcurrido ha sido de 11,68 segundos y en la imagen interpretada se observa en color rojo las variaciones detectadas pero no se muestran las posibles mejoras.

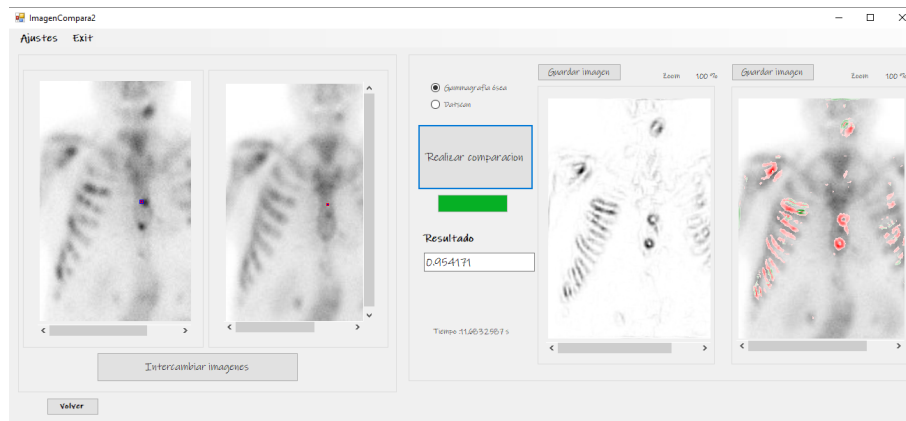


Figura 6.5: Comparación $W=1$ y $C=1$ de zona focalizada, imagen 200 %

En segundo lugar, si se establecen valores de W y C relativamente altos, tales como $W=11$ y $C=11$, se observa la imagen de la figura 6.6 en la cual el tiempo transcurrido aumenta notablemente, siendo 33,16 segundos. En el resultado obtenido se observa una mejor interpretación con respecto a los píxeles verdes que en la figura anterior pero sigue sin ser un resultado óptimo. Además, el tiempo transcurrido ha sido triplicado con respecto al caso anterior.

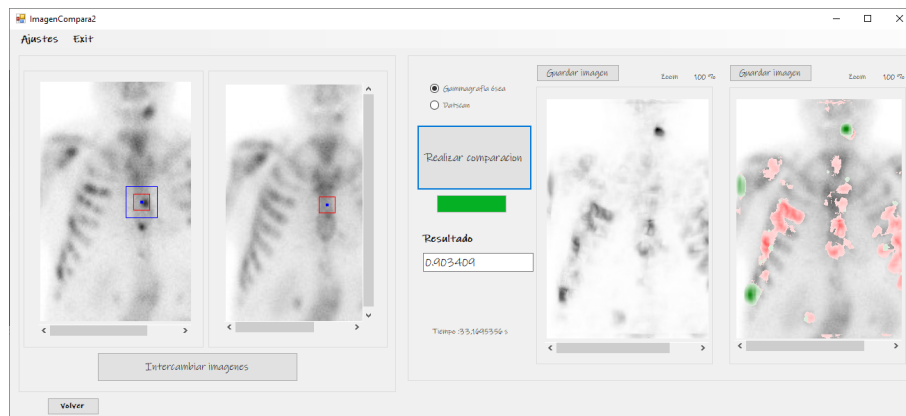


Figura 6.6: Comparación $W=11$ y $C=11$ de zona focalizada, imagen 200 %

En tercer lugar, si se establecen valores de W menores a C , tales como $W=2$ y $C=13$, se obtiene el resultado de la figura 6.7. Se puede observar que el tiempo transcurrido ha sido de 15,06 segundos y el resultado de la imagen interpretada muestra una mejora más aproximada. Las mejoras de las costillas si han sido reflejadas pero la mejora presente en el esternón sigue siendo negativa.

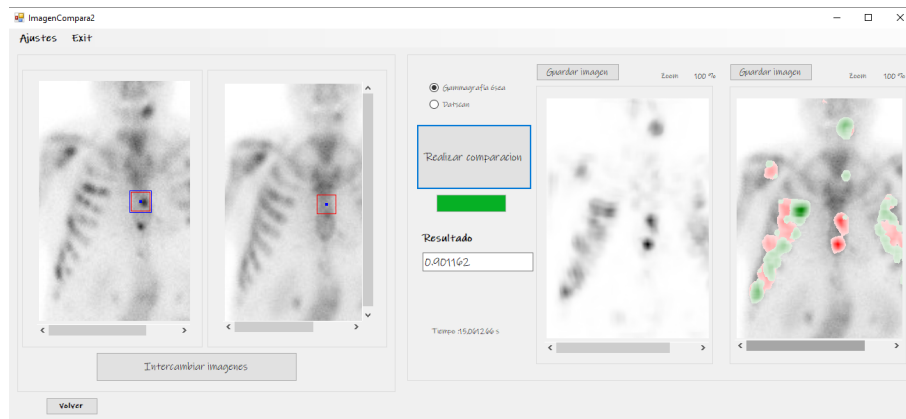


Figura 6.7: Comparación $W=2$ y $C=13$ de zona focalizada, imagen 200 %

Por último, si se establecen valores de W mayores a C , tales como $W=15$ y $C=5$, se obtiene el resultado la figura 6.8, en la cual se puede observar que se realiza una peor interpretación del resultado con respecto a las pruebas anteriores dado que los píxeles verdes correspondientes a una posible mejora han disminuido.



Figura 6.8: Comparación $W=15$ y $C=5$ de zona focalizada, imagen 200 %

Dado que se han obtenido mejores resultados más próximos a nuestra interpretación a priori con valores de W menores que C , se ha realizado una comparación focalizando la zona del esternón en la cual no aparece dicha mejora. Para ello, se ha realizado un zoom sobre ambas imágenes al 300 % y se ha establecido un rectángulo de recorte focalizando dicha zona de 95 píxeles de ancho y 171 píxeles de alto tal y como se puede observar en la figura 6.9. El resultado presente en dicha figura ha sido obtenido con $W=2$ y $C=11$ y se puede observar cómo en este caso, si se muestran correctamente una aproximación de la variación de los píxeles de mejora.

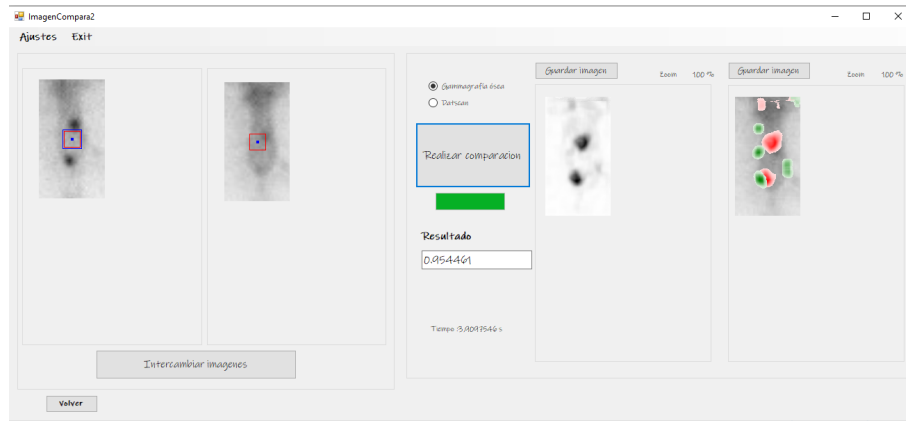


Figura 6.9: Comparación $W=2$ y $C=11$ de zona columna focalizada, imagen 300 %

Por tanto, se puede concluir que en este caso, el resultado óptimo se obtiene con valores de W menores a C y tras realizar un recorte focalizado de las áreas de interés. En concreto se han obtenido mejores resultados con el valor de W igual a 2 y el valor de C próximo a 13.

CAPÍTULO 7

Conclusiones

Una vez finalizada la implementación de la aplicación, se puede concluir que se han alcanzado los objetivos establecidos inicialmente. Gracias a las plantillas utilizadas se ha conseguido crear una interfaz gráfica fácil de usar e intuitiva dividida en diversas ventanas que permiten al usuario seguir el flujo de la aplicación.

Además, se ha conseguido utilizar el algoritmo eficiente BIDM para realizar la comparación entre las imágenes recortadas en la aplicación tras realizar un estudio sobre los diferentes métodos existentes en la literatura para el reconocimiento automático de imágenes y tras solucionar los errores de ejecución correspondientes al formato de las imágenes a comparar.

Asimismo, dependiendo de los parámetros asignados al algoritmo de comparación y de las dimensiones y el área de las imágenes a comparar, se puede obtener un resultado acorde con la evolución del paciente. No obstante, la asignación de los parámetros y dimensiones adecuadas es una tarea compleja que depende de las imágenes a comparar y se deben establecer los parámetros adecuados para obtener un resultado acorde con la evolución real.

Se puede concluir que se obtienen mejores resultados con valores de W y C relativamente bajos con respecto a las dimensiones de las imágenes a comparar y en nuestra herramienta, si se focaliza agrandando la imagen sobre un área concreto de interés, se obtiene un mejor resultado. Además, con valores de W y C bajos, el coste temporal es significativamente menor.

Como ya se ha mencionado anteriormente, la ejecución del algoritmo no es biyectiva, es decir, dadas dos imágenes A y B , no se obtiene el mismo resultado al comparar la imagen A contra la imagen B y viceversa. Por tanto, la elección de la imagen a comparar y la imagen de referencia también condiciona el resultado obtenido.

Con respecto el objetivo de ayudar al personal sanitario a interpretar el resultado, se puede decir que se ha cumplido dado que a través de la imagen interpretada el usuario puede determinar la evolución del paciente según el color de los píxeles asociados. Cabe recordar que la interpretación corresponde con una aproximación que interpreta los píxeles que pueden haber mejorado o empeorado según el estudio realizado, pero no asegura que dicha mejora o empeoramiento sea cierto. Asimismo, para obtener un resultado óptimo dadas dos imágenes A y B a comparar, tal y como se ha podido observar en el capítulo anterior, se deben reajustar los parámetros del algoritmo según la variabilidad entre ambas, además de tener en cuenta sus dimensiones asociadas. Debido a esto, si el personal sanitario no tiene mucha experiencia utilizando nuestra aplicación, puede darse el caso de que no se asignen los parámetros que conducirían a un resultado óptimo o incluso que dichos parámetros lleven a un resultado «erróneo» y por tanto, a una mala interpretación.

Si dentro de dicho objetivo se tiene en cuenta, que además de la interpretación de la imagen resultado, también se debe interpretar el valor obtenido correspondiente a la semejanza entre ambas imágenes, entonces dicho objetivo no se habría cumplido en su totalidad. Esto es debido a que no se puede asociar si el diagnóstico del paciente es favorable o no dado dicho valor. Por ejemplo, si tras la comparación de cuatro imágenes se obtienen los resultados de 0.80 y 0.95, se puede asegurar que existe una diferencia mayor entre las imágenes de la primera comparación con respecto a la segunda comparación, pero en ningún caso, en la implementación actual, puede asociarse dicha variación a un diagnóstico. Para ello, se debe observar la imagen resultado obtenida con los píxeles de variación y la imagen interpretada.

Actualmente, la aplicación de este trabajo se encuentra en fase de pruebas. Pero dado que esta herramienta ha despertado el interés de los médicos del Hospital Universitario Doctor Peset, se pretende continuar para facilitarles una versión final que puedan utilizar en su día a día para ayudarles a determinar la evolución de sus pacientes.

En futuros trabajos, se podrían llevar a cabo diversas mejoras, que en esta versión no han podido ser incorporadas por restricciones temporales, pero que resolverían varios problemas comentados anteriormente.

En primer lugar, se podría incorporar la opción de realizar de forma simultánea la comparación de la imagen A contra la imagen B como referencia y viceversa, realizando la unión y consecuente interpretación de las variaciones obtenidas.

Además, para poder realizar una interpretación del valor obtenido en la comparación, se podría entrenar un modelo asociando el valor de similitud obtenido entre ambas imágenes y el porcentaje de píxeles de cada color obtenidos en el resultado junto con su correspondiente diagnóstico para dado un valor de similitud asociado y un porcentaje de píxeles rojos y verdes correspondientes obtener si nos encontramos ante un caso de evolución favorable o no.

También, otra mejora, podría ser el entrenamiento de otro modelo para que dadas un par de imágenes, las reconozca, informe al usuario sobre el grado de variabilidad de ambas y sugiera al usuario cuales pueden ser los valores de ajuste del algoritmo para obtener un resultado óptimo. De esta forma, se puede ayudar al personal sanitario a reducir el tiempo de ajuste para establecer los parámetros óptimos, además de orientar correctamente a los usuarios no expertos de nuestra aplicación para obtener mejores resultados.

Por último, otra posible mejora sería ampliar los tipos de estudios de medicina nuclear asociados a las imágenes de la comparación para que otras imágenes asociadas a otros estudios tales como gammagrafías tiroideas, entre otras, puedan ser reconocidas y el personal sanitario pueda obtener la ayuda necesaria para determinar la evolución del paciente.

Bibliografía

- [1] Álvaro, F., Sanchez, J.A. y Benedí, J.M. (2013) An image-based measure for evaluation of mathematical expression recognition In *Iberian Conference on Pattern Recognition and Image Analysis*. Springer, Berlin, Heidelberg, 2013. p.682-690.
- [2] Keysers, D., Deselaers, T., Gollan, C. y Ney, H.(2007) Deformation Models for Image Recognition. *IEEE transactions on pattern analysis and machine intelligence*, 29. 1422-35, 2007
- [3] Otsu, N.(1979) A Threshold Selection Method from Gray-level Histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62-66,1979
- [4] Toselli, A.H., Juan, A. y Vidal, E. (2004) Spontaneous Handwriting Recognition and Classification. In *Proceedings of ICPR*, pages 433-436, England, UK,2004
- [5] Liang, L.R., y Loonewy, C.G. (2020) *Dive into Image Processing: Book 1* [Libro electrónico]. N/A Recuperado de https://books.google.es/books?id=WyLdDwAAQBAJ&pg=PP12&lpg=PP12&dq=pgm+image+p2+y+p5&source=bl&ots=Duk1cR8hb8&sig=ACfU3U0w5vYs1iJC5Czr_r0JruqyDr3oug&hl=es&sa=X&ved=2ahUKEwj2a7slanqAhUrxIUKHUc1D1cQ6AEwB3oECAoQAQ#v=onepage&q=pgm%20image%20p2%20y%20p5&f=false
- [6] American Cancer Society.(s.f). *Estudios de medicina nuclear y el cáncer*. Recuperado el 28 de junio de 2020 de <https://www.cancer.org/es/tratamiento/como-comprender-su-diagnostico/pruebas/estudios-de-medicina-nuclear-y-el-cancer.html#:~:text=Algunos%20de%20los%20estudios%20de,Gammagraf%C3%ADa%20 tiroidea>
- [7] Grupo Affidea.(s.f). *Medicina Nuclear*. Affidea. Líderes en diagnóstico por la imagen. Recuperado el 28 de junio de 2020 de https://affidea.es/servicios/pruebas_diagnosticas/medicina-nuclear
- [8] *ImageFormat Class (System.Drawing.Imaging)* (s.f.) Microsoft Docs. Recuperado de <https://docs.microsoft.com/es-es/dotnet/api/system.drawing.imaging.imageformat?view=dotnet-plat-ext-3.1>
- [9] The ImageMagick Development Team.(2020). ImageMagick, Disponible en <https://imagemagick.org>
- [10] Imagen médica. (2020, 3 marzo). En *Wikipedia, la enciclopedia libre*. https://es.wikipedia.org/wiki/Imagen_m%C3%A9dica
- [11] MedlinePlus en español [en línea]. Bethesda(MD): Biblioteca Nacional de Medicina (EE.UU).[actualizado 21 mayo 2020]. Diagnóstico por imágenes. Recuperado el 28 de junio de 2020 de <https://medlineplus.gov/spanish/diagnosticimaging.html>

- [12] MedlinePlus en español [en línea]. Bethesda (MD): Biblioteca Nacional de Medicina (EE.UU). [actualizado 02 junio 2020]. *Gammagrafía ósea*. Recuperado el 28 de junio de 2020 de <https://medlineplus.gov/spanish/ency/article/003833.htm>
- [13] *Online Mockup, Wireframe y UI Prototyping Tool. Moqups*. Recuperado 15 de junio de 2020, de <https://moqups.com>
- [14] *OpenFileDialog Clase (System.Windows.Forms)*. (s.f). Microsoft Docs. Recuperado de <https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.openfiledialog?view=netcore-3.1>
- [15] *Pen Clase (System.Drawing)* (s.f.) Microsoft Docs. Recuperado de <https://docs.microsoft.com/es-es/dotnet/api/system.drawing.pen?view=dotnet-plat-ext-3.1>
- [16] Poskanzer, J. (1991). *PGM Format Specification* <http://netpbm.sourceforge.net/doc/pgm.html>
- [17] *Process Clase (System.Diagnostics)*. (s.f). Microsoft Docs. Recuperado de <https://docs.microsoft.com/es-es/dotnet/api/system.diagnostics.process?view=netcore-3.1>
- [18] *SaveFileDialog Clase (System.Windows.Forms)* (s.f). Microsoft Docs. Recuperado de <https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.savefiledialog?view=netcore-3.1>
- [19] Tomografía computarizada en emisión monofotónica. (2020, 24 junio). En *Wikipedia, la enciclopedia libre*. Recuperado el 29 de junio de 2020 de https://es.wikipedia.org/wiki/Tomograf%C3%ADa_computarizada_de_emisi%C3%B3n_monofot%C3%B3nica
- [20] *Windows Forms*, 30 marzo, 2017. Recuperado 12 de febrero de 2020, de <https://docs.microsoft.com/es-es/dotnet/framework/winforms/>

APÉNDICE A

Instalación de la aplicación

Tras realizar la compilación y publicación del proyecto, se genera el archivo ejecutable de la aplicación para que todos los usuarios puedan ejecutarla. En la figura A.1 se pueden observar todos los archivos generados. Para llevar a cabo la instalación, se debe copiar dicha carpeta en un directorio del equipo donde se quiera instalar la aplicación y tras pulsar sobre el archivo de «setup» se iniciarán los pasos de instalación.

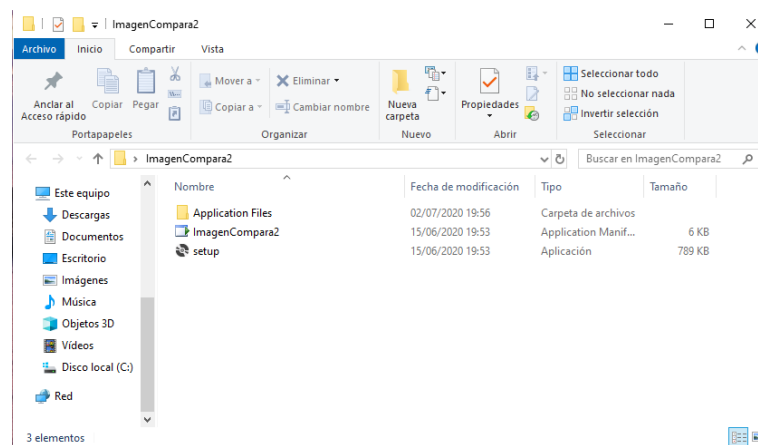


Figura A.1: Carpeta ImagenCompara2

En primer lugar, se comprueba si en el equipo actual se encuentra instalado el *framework* .NET y en caso contrario, instala la versión necesaria para la ejecución de la herramienta. En segundo lugar, aparece la ventana de la figura A.2, mediante la cual el usuario autoriza a que se lleve a cabo la instalación de ImagenCompara2.

Una vez, dicha aplicación ha sido instalada, se puede acceder a ella a través de Programas -> ImagenCompara2 o clicar sobre el archivo ImagenCompara2 que se encuentra en la carpeta de la figura A.1.

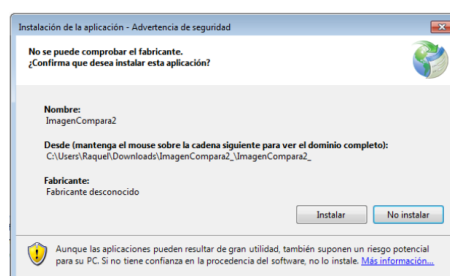


Figura A.2: Ventana de instalación