Centro de Investigación y de Estudios Avanzados
del Instituto Politécnico Nacional

Unidad Zacatenco

**Departamento de Computación**

# Marca de agua para imágenes en dispositivos móviles

Tesis que presenta

**Cynthia Palma Hernández**

para obtener el Grado de

**Maestra en Ciencias**

En la Especialidad de

**Computación**

Directores de Tesis

**Dr. César Torres Huitzil**

**Dr. Adriano de Luca Pennacchia**

México, D.F. Diciembre 2011

La tesis presentada por Cynthia Palma Hernández fue aprobada por:

---

Dr. Guillermo Morales Luna

Dr. Héctor Domínguez Aguirre

Dr. César Torres Huitzil, Director

Dr. Adriano de Luca Pennacchia, Co-Director

México, D.F., 13 de Diciembre de 2011

# Agradecimientos

Sin duda alguna, este trabajo no hubiera sido posible sin el apoyo, amor y comprensión de muchas personas a mi alrededor.

Primero, quiero agradecer a mi familia: a mis padres por todo el soporte que me han dado durante toda mi vida, porque gracias a sus enseñanzas he logrado llegar hasta aquí, es un orgullo ser su hija. A mis hermanas, por todo su cariño, ayuda y tolerancia en los momentos difíciles. Los amo mucho a todos.

A Javi, por su amor incondicional, por su comprensión, por enseñarme que siempre hay una solución, por su ayuda en todo momento y por saber cómo poner siempre una sonrisa en mi cara. Te amo.

A mi asesor Dr. César Torres, gracias por todo el aprendizaje, por su apoyo y comprensión. Gracias por haber confiado en mí.

A mis sinodales, Dr. Guillermo Morales y Dr. Héctor Domínguez, por haberse tomado el tiempo de leer mi trabajo y por todas las críticas constructivas que me aportaron. Gracias.

A Susa, Joana, Eliza, Yadira, Sandy y Sayra, porque después de casi ocho años seguimos siendo tan amigas, confidentes y cómplices como siempre. Las quiero mucho.

Por supuesto también agradezco a todos los que de alguna u otra manera contribuyeron para que este día llegara y para que esta tesis tuviera su fin. A mis compañeros de generación, a mis profesores, a Sofi y a Lau. Gracias por esa ayuda desinteresada.

Al CONACyT por haberme otorgado la beca. Gracias.

# Índice general

# Índice de figuras

# Índice de cuadros

# Lista de Acrónimos

**DRM** del inglés *Digital Rights Management*

**EOF** del inglés *End Of File*

**HVS** del inglés *Human Visual System*

**SNR** del inglés *Signal-to-Noise Ratio*

**PSNR** del inglés *Peak Signal-to-Noise Ratio*

**DWT** del inglés *Discrete Wavelet Transform*

**DCT** del inglés *Discrete Cosine Transform*

**NIST** del inglés *National Institute of Standards and Technology*

**FPGA** del inglés *Fiel Programmable Gate Array*

**GPU** del inglés *Graphics Processing Unit*

**IC** del inglés *Integrated Circuit*

**DCS** del inglés *DCT Coefficients Selection*

**LFSR** del inglés *Linear Feedback Shift Register*

**NC** del inglés *Normalized Correlation*

**MSE** del inglés *Mean Square Error*

**SDK** del inglés *Software Development Kit*

**API** del inglés *Application Programming Interface*

**XML** del inglés *eXtensible Markup Language*

**NDK** del inglés *Native Development Kit*

**JNI** del inglés *Java Native Interface*

**DFT** del inglés *Discrete Fourier Transform*

**Resumen**

Actualmente los dispositivos que utilizamos día con día para el manejo de información digital van más allá de la computadora personal o laptop. En los últimos años el uso de dispositivos móviles se ha incrementado muy rápidamente y debido a su ubicuidad el tráfico de documentos digitales multimedia también se ha multiplicado, por esta razón es importante contar con sistemas que nos permitan gestionar tanto la creación como la distribución de contenido digital, ya que por su naturaleza son facilimente reproducidos, copiados y modificados.

Así pues, podemos decir que la administración de derechos de autor (del inglés *Digital Rights Management* (DRM)) para contenido digital es un conjunto de tecnologías usadas para controlar tanto el acceso a la información como el uso de la misma, aunque también debemos considerar el problema de autenticación de dicho contenido digital. En el presente trabajo se aborda dicha problemática enfocada principalmente a dispositivos móviles.

Las marcas de agua son una herramienta basada en antiguas técnicas y mecanismos esteganográficos para el ocultamiento de la información, así que su uso como sistema de protección proviene de tiempo atrás. El uso de esquemas de marca de agua constituyen un mecanismo alternativo o complementario a otros que existen actualmente como: la criptografía.

Una marca de agua es una técnica de ocultamiento de información dentro de un objeto, existen diversas clasificaciones para los esquemas de marca de agua. En este trabajo nos enfocamos principalmente en dos tipos de esquemas de marca de agua: un esquema de marcado robusto que es utilizado para la protección de derechos de autor y un esquema de marcado frágil que se utiliza para la autenticación de imágenes.

Tomando en cuenta que el principal desafío de este trabajo es la implementación de los esquemas de marcado de agua en un dispositivo móvil, se describen algunos procedimientos que mejoran la eficiencia de los esquemas, ya que un dispositivo móvil tiene restricciones en cuanto a capacidad de procesamiento y almacenamiento.

Se realizaron distintas pruebas a los esquemas de marcado de agua para comprobar y verificar su desempeño contra distintos ataques como son: geométricos, de interferencia, de compresión con pérdida, etc. Finalmente se presenta un aplicación que se desarrolló utilizando los procedimientos descritos y que tiene como finalidad mostrar que es posible implementar en ambientes restringidos algoritmos que tienen un costo computacional considerable.

Palabras clave: marcas de agua, imágenes, dispositivos móviles, Android.

**Abstract**

The fast development of electronic devices and proliferation of the Internet lead to an exponential growth of multimedia document traffic (image, text, audio, video, etc.). This phenomenon is now so important that insuring protection and control of the exchanged data has become a major issue. Indeed, due to their digital nature, multimedia documents can be duplicated, modified, transformed, and distributed very easily. In this context, it is important to develop systems to manage the creation and distribution of digital content.

Digital Rights Management (DRM) refers to a collection of technologies used to control the access and use of digital data, it is generally used for copyright protection. However, we must not forget that there are other security services, such as image content authentication. In this work, we take into account these security services focused on mobile devices.

Watermarking seems to be a complementary solution to other security mechanisms as cryptography for image integrity and authenticity . The aim of watermarking is to include subliminal information (i.e., imperceptible) in a multimedia document to ensure a security service and generates a watermarked multimedia object, which can be an image, audio, video or text. There are many watermarking techniques, these techniques can be classified according to a number of different criteria. In this work, we focused on two types of watermarking schemes: a robust watermarking scheme which is used for copyright protection and a fragile one that is used for image authentication.

The main purpose of this work is to show that now is possible to construct watermarking schemes on mobile devices. We present an implementation on a mobile device of the two watermarking schemes mentioned before. It is well know that mobile devices have lower memory capacity and processing power than desktop computers. Therefore mobile applications must be designed to optimize the use of data storage and processing power, these features are needed to create efficient software systems. Hence, there were some adjustments to improve the efficiency of watermarking schemes.

A series of experiments was conducted to verify the performance of the watermarking schemes. These experiments involve attacking the watermarked images using different techniques like: geometric attacks, addition of noise and lossy compression, etc. Finally, an Android application is presented, which was developed to test the correctness of the two implemented watermarking schemes. Also, with this application we show that is possible to build mobile applications that use computationally expensive operations.

Key words: watermarking, images, mobile devices, Android.

# Capítulo 1

# Introducción

## 1.1. Antecedentes y contexto de la investigación

Con los grandes avances de la tecnología, el tipo de dispositivos para el manejo de información digital que estamos acostumbrados a utilizar ha evolucionado considerablemente. Hace algunos años el manejo de información digital se hacía únicamente en computadoras personales de escritorio.

Actualmente el uso de dispositivos móviles se ha incrementado rápidamente lo cual hace que el manejo de información digital (imágenes, audio, vídeo y texto) sea una actividad cotidiana para cualquier persona. La ubicuidad de los dispositivos móviles ha generado un crecimiento exponencial en el tráfico de documentos digitales multimedia, este fenómeno es ahora tan importante que asegurar la protección y el control de los datos intercambiados se ha convertido en un tema de gran interés debido a la facilidad con la que los datos digitales pueden ser reproducidos, copiados y modificados [5]. Por lo tanto, en los ultimos años se ha impulsado la búsqueda de nuevas alternativas para enfrentar efectivamente el uso y la manipulación ilegal de dicha información.

La administración de derechos digitales (DRM), se refiere a un conjunto de tecnologías usadas para controlar el acceso a datos digitales [1]. Dentro de este conjunto podemos encontrar técnicas, procedimientos y algoritmos relacionados con el establecimiento de un entorno informático de confianza y una infraestructura confiable para la seguridad en la creación, la transportación y la prevención del uso indebido y/o el consumo de contenidos digitales protegidos.

Dentro del contexto de la DRM, las marcas de agua constituyen un mecanismo alternativo o complementario a otros que existen actualmente tal como: la criptografía. Las marcas de agua surgen como una respuesta a los retos que impone la administración de derechos de autor ya que proporcionan la certeza de saber que, no importa cómo o dónde aparezcan, los documentos digitales llevan el aviso de propiedad [2].

Con el paso del tiempo las características de los dispositivos móviles han mejorado

considerablemente, ahora son más pequeños y tienen más capacidad de procesamiento, además de que tienen otro tipo de aditamentos que complementan su funcionamiento, tal es el caso de la cámara fotográfica. Debido a estas características, se puede catalogar a un dispositivo móvil como productor y consumidor de imágenes digitales por lo que otro factor importante que se debe considerar en cuestiones de seguridad es la autenticidad e integridad de la mismas.

Los servicios de seguridad antes mencionados, pueden lograrse mediante el uso de diferentes esquemas de marcas de agua, de tal forma que cuando alguien tome un fotografía digital tenga la opción de protegerla para que pueda intercambiarse y viajar a través de canales de comunicación que no necesariamente sean seguros. Y que de igual forma, cuando alguien recibe una imagen digital pueda comprobar el origen de la misma o bien verificar que el contenido no haya sido modificado. Algunos ejemplos de áreas de aplicación para los esquemas de marca de agua son: el comercio electrónico (e-commerce) y el gobierno electrónico (e-governance), en dónde es necesario proteger los datos digitales que son intercambiados [18].

Las marcas de agua son una herramienta basada en antiguas técnicas y mecanismos esteganográficos para el ocultamiento de la información, así que su uso como sistema de protección proviene de tiempo atrás. En 1945 se ocultó código Morse en un dibujo (ver Figura 1.1), la información oculta está codificada en el tramo de hierba a la orilla del río [6].

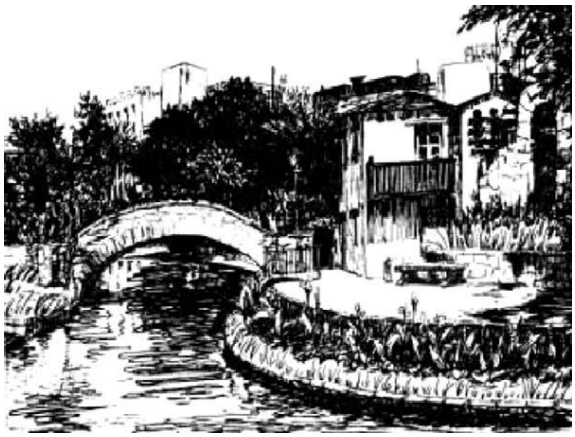


Figura 1.1: Código Morse oculto en la hierba a la orilla del río

Una marca de agua es una técnica de ocultamiento de información dentro de un objeto que puede ser una imagen, audio, video o texto. Existen dos enfoques para los esquemas de marcas de agua: uno de ellos es en el dominio espacial y el segundo es en el dominio de la frecuencia o transformado, aunque existen muchas más clasificaciones, ésta es una de las más importantes. En general, una marca de agua digital debe cumplir con las siguientes características, aunque éstas pueden variar según el uso que se le esté dando: visible o imperceptible, que no degrade el objeto, robusta o frágil, no debe ser ambigua, entre otras.

Con el desarrollo de los sistemas de cómputo y el inicio de la era digital el uso de marcas de agua ha tomado un gran auge. Actualmente existen varios productos comerciales que utilizan diferentes esquemas de marcado de agua. Podemos encontrar implementaciones muy sencillas que se valen de la etiqueta de fin de archivo (EOF) para hacer la inserción de la información a ocultar. La idea detrás de este enfoque es abusar del reconocimiento de la etiqueta EOF, *i.e.*, la información de la marca se inserta después de dicha etiqueta [6], así el objeto no se degrada visualmente pero el grado de seguridad que presenta dicho esquema es muy bajo ya que al visualizar el archivo en cualquier editor de notas la información puede ser facilmente eliminada o modificada (ver Figura 1.2). Existen otros trabajos de investigación que hacen implementaciones con un esquema de marcas de agua más elaborado por lo que su grado de seguridad es mejor, estas implementaciones utilizan algoritmos que trabajan en el dominio espacial o en el dominio de la frecuencia.


Figura 1.2: Información oculta después de la etiqueta EOF de un archivo

En general, un sistema de marcas de agua, requiere de varias etapas desde que se inserta la marca, hasta que es detectada [1]: la primera etapa, que se conoce como proceso de inserción, requiere de la marca de agua y de la imagen original (sin marcar) para dar como resultado una imagen marcada. La siguiente etapa, denominada proceso de detección o extracción, trata de detectar o extraer la marca de agua para finalmente decidir si dicha marca es o no aceptada. Es importante tener en consideración que la imagen que ha sido marcada puede ser utilizada o transportada en ambientes que puedan distorsionar la información que contiene, es decir, está propensa a sufrir cierto tipo de ataques ya sean intencionados o no, la resistencia a los diferentes ataques depende del nivel de robustez del esquema de marcado de agua.

En los últimos años, varios esquemas de marca de agua han sido propuestos, pero los esquemas orientados a dispositivos móviles son escasos. Debido a esto en el presente trabajo se aborda dicha problemática y se decidió implementar dos esquemas de marca de agua: un esquema robusto que nos permita brindar servicios de protección para el contenido digital y un esquema frágil para la autenticación de imágenes. Sin embargo dichos esquemas están diseñados con muy poca o sin ninguna consideración sobre las limitaciones de los dispositivos móviles tales como: complejidad computacional o disponibilidad de energía. Debido a esto el principal desafío consiste en hacer adaptaciones o mejoras al desempeño del esquema (en cuestiones de consumo costo computacional y uso de memoria) sin afectar

su funcionamiento.

En cuanto a servicios de seguridad, hoy en día las marcas de agua han encontrado distintas aplicaciones [3, 4], como pueden ser: comprobar la propiedad de los archivos digitales, autenticar y verificar la integridad de la información, etiquetar el contenido, proteger y controlar el uso de la información, combatir el uso fraudulento de comunicaciones de voz inalámbricas, autenticar la identidad de un teléfono celular o estaciones de transmisión y asegurar la entrega de música y otros contenidos de audio.

## 1.2. Planteamiento del problema

La simplicidad en el procesamiento es una característica deseada en los ambientes de recursos limitados tales como dispositivos móviles, sin embargo la mayoría de los métodos de marcado de agua no han sido diseñados para este tipo de plataformas. La implementación de procedimientos nos ayuden a la administración de derechos de autor y a garantizar la autenticidad e integridad de imágenes digitales utilizando marcas de agua en plataformas móviles, es particularmente interesante debido a que las capacidades de procesamiento y energía de dichos dispositivos representan la mayor limitante, ya que generalmente los algoritmos de marcas de agua hacen uso de transformaciones, matrices, números reales, números pseudo-aleatorios, etc.

Actualmente, es poco el trabajo que se ha elaborado en esta área de investigación, no se sabe de forma certera si existen esquemas de marca de agua que sean totalmente eficientes y que puedan ser implementados en dispositivos móviles. La posibilidad que encontramos es la de hacer algún tipo de modificación o adaptación que nos permita reducir el costo computacional de los esquemas de marcado. Sin embargo, ésta posibilidad nos hace plantear un compromiso entre las ventajas y desventajas que ésto representa en cuanto a eficiencia, cuestiones de seguridad y administración de los recursos disponibles.

Recientemente las implementaciones en hardware comienzan a ser más comunes puesto que son esenciales para lograr un bajo consumo de energía, funcionamiento en tiempo real y con alta confiabilidad, entre otras características. Sin embargo, muchas veces este tipo de sistemas no es de fácil integración, *i.e.*, aunque se creara un sistema embebido de marca de agua que fuera eficiente, integrarlo en la fabricación de dispositivos móviles de las grandes compañías significaría el mayor reto.

Es por esto que en el presente trabajo de tesis se pretende abordar dicha problemática desde un enfoque de desarrollo de software, *i.e.*, llevar a cabo la implementación de diferentes algoritmos de marcas de agua que funcionen de forma eficiente y hagan un uso adecuado de los recursos del dispositivo, es decir, que tengan el menor costo computacional posible y que hagan un buen uso de los recursos de memoria, ya que algunos de los algoritmos utilizan estructuras de datos complejas como matrices y también requieren el

cálculo de transformaciones.

## 1.3. Objetivos del trabajo de investigación

### 1.3.1. Objetivo general

- Construir un esquema que brinde servicios de protección para la información digital mediante el uso de marcas de agua para imágenes sobre plataformas móviles.

En este trabajo de investigación se plantea utilizar principalmente algoritmos invisibles de marcado de agua para imágenes, reduciendo el alcance del trabajo a los siguientes aspectos: protección de derechos de autor mediante un algoritmo robusto y autenticación de la información mediante un algoritmo frágil. Se desea que dichos algoritmos sean eficientes y hagan un buen uso de los recursos del dispositivo móvil.

### 1.3.2. Objetivos específicos

- Investigar métodos de marcado de agua convenientes para dispositivos móviles.
- Establecer un esquema de autenticación para imágenes digitales utilizando un *algoritmo de marcado de agua frágil e invisible* sobre dispositivos móviles.
- Emplear un *esquema de marca de agua robusto e invisible* para la protección de derechos de autor en imágenes digitales sobre dispositivos móviles.
- Elaborar una implementación eficiente de los esquemas de marcas de agua en plataformas móviles, con el fin de lograr un uso óptimo de los recursos disponibles.
- Validar los esquemas de marcas de agua seleccionados mediante la implementación de una aplicación en un dispositivo móvil.

## 1.4. Organización de la tesis

La tesis consta de seis capítulos, los cuales están organizados de la siguiente manera:

El capítulo 2 trata del marco general del marcado de agua, así como la revisión del estado del arte. Se presenta la definición y los conceptos básicos del marcado de agua: clasificación, distintas aplicaciones y posibles ataques. Se mencionan también las características deseables en un esquema de marcado de agua y a partir de estas características se destacan, mediante un estudio comparativo, las principales ventajas y desventajas de los algoritmos estudiados.

Los capítulos 3, 4 y 5 contienen la información que refleja los principales aportes de este trabajo de investigación. En el capítulo 3 se presentan y describen cada una de las etapas que conforman los esquemas de marcado de agua que fueron estudiados. En la primera sección de este capítulo se expone el esquema frágil de marcado de agua descrito en [1] que está basado en la teoría del caos, se explica su funcionamiento, ventajas y desventajas. Dentro de la segunda sección se presenta el esquema robusto de marcas de agua y se mencionan sus principales características de funcionamiento, alcances y limitaciones.

En el capítulo 4 se describen las estrategias que se emplearán para hacer que los algoritmos seleccionados sean lo más eficientemente posibles así como los resultados obtenidos de realizar distintas pruebas a cada uno de los esquemas de marcado de agua, dichas pruebas consisten medir el desempeño de los esquemas de marcado ante diferentes ataques. Posteriormente, en el capítulo 5 se presenta la implementación de una aplicación de prueba sobre Android, que se construyó para verificar el funcionamiento de los esquemas previamente elegidos, en esta sección se encuentran aspectos de diseño y los tiempos de ejecución obtenidos de probar la aplicación en un dispositivo móvil en específico.

Finalmente, en el capítulo 6 se presentan las conclusiones acerca del desarrollo de este trabajo, así como algunas ideas de trabajo a futuro que podrían mejorar el funcionamiento de los esquemas de marcado de agua que se presentaron.

# Capítulo 2

# Marco general de las marcas de agua y estado del arte

Las imágenes y en general cualquier objeto multimedia pueden ser más de lo que podemos ver a través de nuestro Sistema Visual Humano (del inglés *Human Visual System* (HVS)). Durante décadas las personas se esforzaron por desarrollar métodos innovadores que permitieran establecer una comunicación secreta y que al mismo tiempo protegieran cierto tipo de información que se consideraba importante.

Existen varias disciplinas que nos ayudan a lograr este fin y que están estrechamente ligadas: esteganografía, marcado de agua y criptografía [3], como se puede apreciar en la Figura 2.1. Las dos primeras son muy difíciles de separar ya que ambas describen métodos para insertar información dentro de algún objeto portador. Por un lado, la esteganografía esconde información independientemente del objeto que se utilice para insertar los datos y debe ser estadísticamente indetectable. En cambio en las marcas de agua, es de importancia el objeto en donde se insertarán los datos a ocultar y normalmente es de conocimiento público que el objeto contiene una marca.

A continuación se presenta el marco general de las marcas de agua, su definición, los diferentes tipos de esquemas de marcado de agua que podemos encontrar en la literatura y sus principales aplicaciones. Además de esto, se mencionan también varios ataques a los que son susceptibles los objetos marcados ya que cuando se diseña un esquema de marcado de agua es importante tenerlos en cuenta.

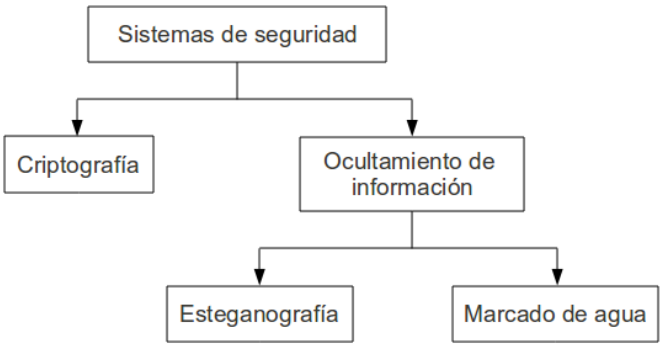



Figura 2.1: Las diferentes disciplinas que tratan la seguridad de los sistemas

## 2.1. Marcas de agua

Dentro del siguiente apartado se describen los conceptos y fundamentos básicos que son necesarios para tener un buen entendimiento de cualquier esquema de marcado de agua. Se presenta la definición así como las partes esenciales de cualquier esquema de marcado de agua, se expone también una clasificación de acuerdo a los aspectos más generales que caracterizan un esquema de marcado de agua, finalmente se mencionan las aplicaciones y ataques a los que pueden ser sometidos.

### 2.1.1. Definición

En general, el marcado de agua es el proceso de insertar alguna información, llamada marca de agua, dentro de un objeto multimedia, de tal forma que posteriormente dicha marca de agua pueda ser detectada o extraída para hacer una afirmación acerca de dicho objeto que puede ser una imagen, audio, video o texto.

Cualquier esquema de marcas de agua está formado por distintas partes [3]: *la marca de agua, el codificador, el decodificador y el comparador*. A continuación se define formalmente cada una de ellas:

- *La marca de agua (W)*: es usualmente una imagen en el caso de esquemas de marcas de agua visibles. Para el caso de marcas de agua invisibles puede utilizarse una imagen binaria o un número seudoaleatorio. En algunos esquemas se maneja una llave privada que generalmente se utiliza como semilla de un generador de números seudoaleatorios

- *El codificador (E)*: Corresponde al proceso de inserción de la marca de agua, formalmente se puede definir como una función $E$ que toma una imagen $I$, una marca de agua $W = \{w_1, w_2, \ldots\}$ y genera una imagen marcada $\hat{I}$, lo que se puede expresar con la siguiente expresión(2.1):

$$E(I, W) = \hat{I} \tag{2.1}$$

Cabe señalar que la marca de agua $W$ puede depender de varios factores tales como: el tamaño de la imagen original, una llave de usuario, las características propias de la imagen, etc. Aunque en cualquier caso la definición no se afecta.

- *El decodificador (D)*: Corresponde al proceso de extracción o deteccción de la marca de agua, de manera formal puede definirse como una función $D$ que toma una imagen $J$ (la cual puede estar marcada o no, y posiblemente corrupta) y extrae o detecta una marca de agua $W'$. Es importante mencionar que en este proceso, dependiendo de las caracteristicas del esquema de marcado, puede ser necesario o no incluir una imagen adicional $I$, la cual generalmente corresponde a la versión original y sin marcar de $J$, lo anterior puede escribirse como en la ecuación (2.2).

$$D(J, I) = W' \tag{2.2}$$

- *Comparador (C)*: la marca de agua extraída en el proceso de decodificación ($W'$) es comparada con la del propietario mediante la función de comparación $C_\delta$ y se genera como salida una decisión binaria: 1 si hay una coincidencia y 0 de otra forma, tal y como se muestra en la ecuación 2.3.

$$C_\delta(W', W) = \begin{cases} 1, & c \leq \delta \\ 0, & \text{en otro caso} \end{cases} \tag{2.3}$$

Donde, $c = C_\delta(W', W)$ es la correlación entre las dos marcas de agua y $\delta$ es un umbral. Algunas de las métricas más usadas como valores de umbral son el cálculo de la proporción señal-a-ruido (del inglés *Signal-to-Noise Ratio* (SNR)) y la proporción máxima de señal-a-ruido (del inglés *Peak Signal-to-Noise Ratio* (PSNR)).

### 2.1.2. Clasificación

Un gran número de técnicas de marcado de agua están disponibles en la literatura, todas estas técnicas pueden ser agrupadas dentro de diferentes categorías [3, 4, 8], las más importantes son (ver Fig.2.2):

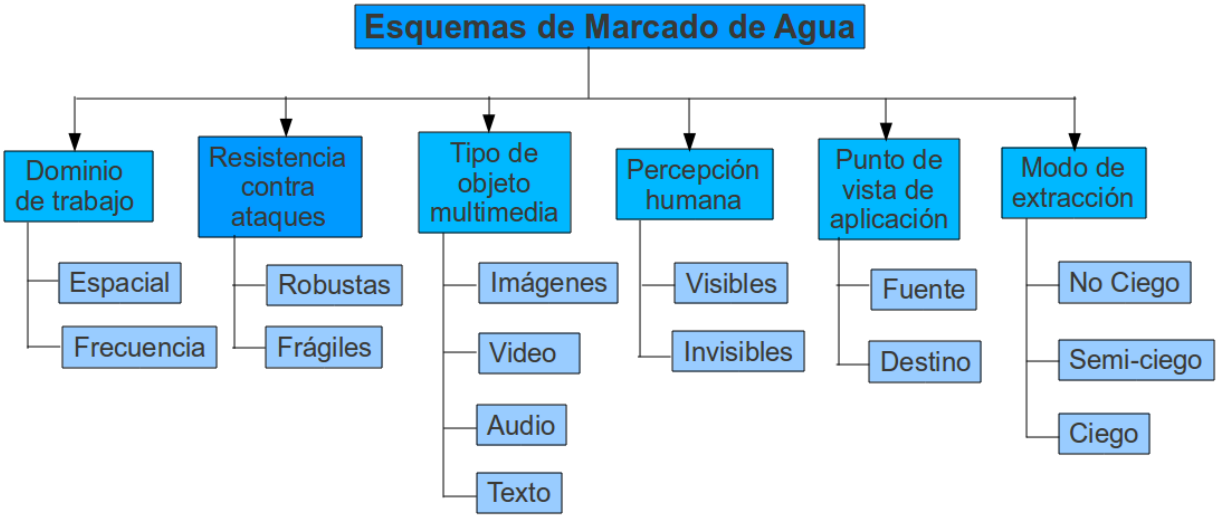



Figura 2.2: Clasificación de los esquemas de marcado de agua

- Según el dominio de trabajo del algoritmo que se utilice durante el proceso de inserción y/o extracción:
  - *Dominio del espacio*: este tipo de algoritmos modifican el valor de un dato(píxel) utilizando la relación que existe con sus vecinos. Son computacionalmente más baratas y fáciles de implementar [2].
  - *Dominio de la frecuencia*: utiliza transformadas que trabajan en el dominio de la frecuencia como: trasformada discreta de Wavelet (del inglés *Discrete Wavelet Transform* (DWT)), transformada discreta del coseno (DCT), entre otras para modificar los coeficientes obtenidos. En general este tipo de enfoque es más robusto contra los ataques o modificaciones que existen (adición de ruido blanco, transformaciones afines, comprensión, rotación, entre otros).
- Según su resistencia contra ataques, ya sean intencionados o accidentales, entendiendo por ataque cualquier procesamiento que debilite o elimine la marca de agua. El desempeño de un algoritmo de marcado de agua contra los diferentes ataques refleja su calidad [3].
  - *Marcas de agua robustas*: pueden diseñarse para soportar un cierto grado de modificación, dependiendo de las necesidades de la aplicación.
  - *Marcas de agua frágiles*: son diseñadas para destruirse o modificarse ante cualquier distorsión sobre la imagen que la contiene.
- Según el tipo de objeto multimedia dentro del cual se insertará la marca, se pueden clasificar como:
  - *Marcas de agua para imágenes*

  - *Marcas de agua para vídeo*
  - *Marcas de agua para audio*
  - *Marcas de agua para texto*

  Es difícil construir un algoritmo de marcas de agua que trabaje de forma efectiva para todos estos objetos ya que cada uno debe tratarse de forma diferente.

- Según la percepción humana las marcas de agua pueden clasificarse como:

  - *Marca de agua visible*: es una imagen secundaria traslucida sobrepuesta en una imagen primaria (imagen original), la marca de agua aparece visible para el observador con una inspección cuidadosa.
  - *Marca de agua invisible*: es completamente imperceptible para el sistema visual humano, este tipo de marcas de agua se utiliza cuando se desea mantener la fidelidad o calidad de la imagen original.

- Desde el punto de vista de una aplicación las marcas de agua pueden ser:

  - *Basadas en la fuente*: este tipo de marca de agua es deseable para identificación de propiedad o autenticación, debido a que una única marca de agua que identifica al propietario es insertada en todas las copias de una objeto multimedia que van a ser distribuidas.
  - *Basadas en el destino*: en este enfoque a cada copia distribuida se le inserta una marca de agua única que identifica a un comprador en particular. Este tipo de marcas de agua podrían ser utilizadas para rastrear al comprador en el caso de reventa ilegal.

- Según los requerimientos del sistema de marcado de agua durante la etapa de extracción/detección se pueden clasificar como:

  - *Sistema de marcado de agua no-ciego*: requiere al menos el objeto multimedia original y de manera opcional la marca de agua.
  - *Sistema de marcado de agua semi-ciego*: no hace uso del objeto multimedia original, este tipo de sistemas son esenciales cuando el acceso a la información original no es posible o es impractico.
  - *Sistema de marcado de agua ciego*: este tipo de sistemas representa un reto mucho mayor, ya que ni el objeto multimedia original ni la marca de agua son requeridos.

### 2.1.3. Aplicaciones

Las aplicaciones de las marcas de agua se han incrementado, y frecuentemente se proponen esquemas que son utilizados en diferentes áreas en [2–4, 7] se mencionan varias, algunas de estas aplicaciones son:

- *Monitoreo de transmisiones*: frecuentemente las empresas pagan el servicio de algún medio de comunicación para que transmita una cierta cantidad de anuncios publicitarios. Para asegurarse de que realmente se transmita la cantidad acordada de publicidad se contrata personal para monitorear los medios de comunicación. Sin embargo, este método es engorroso y muy susceptible a errores. La propuesta es la de utilizar marcas de agua en la señal transmitida por los medios de comunicación, de manera que se pueda colocar una computadora que detecte dichas marcas, y contabilizar de manera automática las transmisiones efectivas. Este tipo de aplicaciones requieren de legislaciones que obliguen a los medios de comunicación a incluir las marcas de agua en su señal.

- *Identificación de propietarios*: usar marcas visibles en imágenes protegidas por los derechos de autor es una práctica muy común. Sin embargo, estas marcas pueden disminuir la calidad de las imágenes, y son susceptibles a ser recortadas, modificadas, etc. La aplicación de las marcas de agua para reforzar los derechos de propiedad, es probablemente la más usual por su característica de invisibilidad, asegurando una alta calidad de las imágenes e incrementando la robustez ante diferentes distorsiones.

- *Monitoreo de distribución*: existen documentos importantes que contienen información confidencial que no debe ser distribuida sin autorización. Se han hecho propuestas para agregar marcas de agua diferentes a todos los datos distribuidos legalmente. De esta forma, se puede detectar el origen de copias rea- lizadas ilícitamente. La robustez e imperceptibilidad de las marcas son de gran ayuda en esta aplicación.

- *Verificación de integridad*: actualmente se dispone una gran cantidad de herramientas de edición de imágenes que permiten modificar su contenido visual, con tanta calidad que resulta difícil detectar los cambios realizados. La criptografía ha estudiado este caso durante mucho tiempo usando firmas digitales. Sin embargo, una vez que se tiene acceso a la firma, la imagen puede ser modificada y distribuida ilegalmente. Existe un grupo de marcas de agua, conocidas como frágiles (ver Sección 2.1.1), que están diseñadas para modificarse con cualquier distorsión sufrida por la imagen marcada, pudiendo verificar fácilmente su integridad.

- *Control de copias*: la distribución de copias ilegales de material protegido por derechos de autor, representa pérdidas millonarias para las compañías que lo producen. Al agregar marcas de agua dentro del material que se desea proteger, y obligando a los fabricantes a incluir un detector de marcas de agua en los reproductores de este material (agregándolo en las patente de los formatos), es posible evitar copias ilegales. Un ejemplo de esto son los sistemas de DVD que contienen información de copia en una marca de agua.

- *Aplicaciones en imágenes médicas*: la utilización de imágenes digitales en la medicina se incrementa frecuentemente, y con ello, la necesidad de garantizar la seguridad en este tipo de información con el fin de:

  - Incrementar la utilidad de las imágenes

- Autenticar las imágenes
- Verificar la integridad de las imágenes

### 2.1.4. Ataques

Un objeto que ha sido marcado está propenso a sufrir ataques, ya sean intencionados o no, antes de que llegue a su destino. Un ataque es cualquier procesamiento que debilita o elimina la marca de agua. El desempeño de un algoritmo de marcas de agua contra este tipo ataques refleja su calidad. Entre los ataques más comunes podemos encontrar:[3].

- *Ataques de eliminación e interferencia*: por un lado están los ataques de eliminación que tratan de quitar la información de la marca de agua que se encuentra dentro de un objeto, por otra parte, los ataques de interferencia son aquellos que añaden ruido adicional al objeto marcado. Dentro de esta categoría podemos encontrar: compresión con pérdida, modulación y tormentas de ruido.

- *Ataques geométricos*: son específicos para imágenes y videos, no quitan la marca de agua, pero manipulan el objeto marcado de tal forma que no se puedan localizar la información de la marca. Este tipo de ataques incluyen transformaciones afines como: rotación, traslación y escalamiento.

- *Ataques criptográficos*: tratan de romper la seguridad del algoritmo.

- *Ataques al protocolo*: este tipo de ataques aprovechan las lagunas del concepto de marca de agua, por ejemplo el ataque IBM: consiste en insertar una o varias marcas de agua de tal forma que es difícil distinguir la original.

- *Ataques físicos*: pueden tomar lugar cuando los sistemas de marcado de agua y su correspondiente administrador de derechos digitales están dentro de un sistema embebido.

Además de estos ataques que son los más generales, podemos encontrar ataques más elaborados, en [5] se describen los siguientes:

- Tratar de modificar la imagen sin alterar la marca de agua insertada.

- Tratar de crear una nueva marca de agua que el autenticador considerare como auténtica

- Cuando la integridad de una imagen se basa en una marca de agua que es independiente del contenido de la imagen es posible desarrollar un ataque que pueda copiar una marca de agua válida dentro de otra imagen, así la segunda será falsa pero pasará como auténtica.

- El ataque conocido como *"Collage Attack"* propuesto por Fridrich [19], crea una imagen falsa utilizando partes de un grupo de imágenes protegidas por la misma marca y la misma llave. Para este ataque no se necesita ningún conocimiento previo acerca de la marca de agua o la llave secreta. Su principio es relativamente fácil ya que consiste en reemplazar cada píxel de la imagen alterada por el valor del píxel más cercano en la imagen base, la principal dificultad de este ataque radica en obtener una base de datos de imágenes que sea suficientemente numerosa para así obtener una imagen falsa de buena calidad visual.

- Otro ataque clásico es tratar de descubrir la llave secreta utilizada para generar la marca de agua. Esta clase de ataque, también llamado ataque de fuerza bruta, es muy conocido dentro de la comunidad de seguridad. Una vez que la llave ha sido encontrada es muy fácil falsificar la marca de agua de una imagen que ha sido protegida por esta llave. La única forma de evitar este ataque es usar llaves largas para disuadir al atacante de tratar de descubrir la llave debido al alto costo computacional.

## 2.2. Revisión del estado de arte

El presente trabajo de tesis está enfocado principalmente a dos servicios de seguridad: comprobación de derechos de autor y autenticación e integridad de la información, dichos servicios estarán enfocados en imágenes digitales. Como se mencionó en las secciones anteriores existen varios tipos de esquemas de marcado de agua, cada uno de ellos posee diferentes características según el uso que se les esté dando. Para el trabajo que se está desarrollando se propone lo siguiente (ver Figura 2.3):

- Para brindar el servicio de comprobación de derechos de autor se plantea trabajar con un esquema de marcado de agua que haga uso de un algoritmo invisible y robusto. Estas son dos características deseables ya que, como se mencionó en la Sección 2.1.2, una marca de agua invisible hace que la calidad y fidelidad de la imagen no se vea degradada, en el caso de ser una imagen comercial, por otro lado una marca de agua robusta se caracteriza por tener una mayor resistencia a diferentes ataques, garantizando hasta cierto punto que el autor de la imagen podra reclamar su propiedad.

- Para el servicio de autenticación e integridad de la imagen se va a hacer uso de un esquema de marcado de agua que utilice un algoritmo invisible y frágil. Como vimos en la Sección 2.1.2, las marcas de agua frágiles son especialmente diseñadas para destruirse o modificarse ante cualquier distorsión sobre la imagen que la contiene, de esta forma es posible detectar cualquier cambio que se haga a la imagen protegida por la marca de agua.

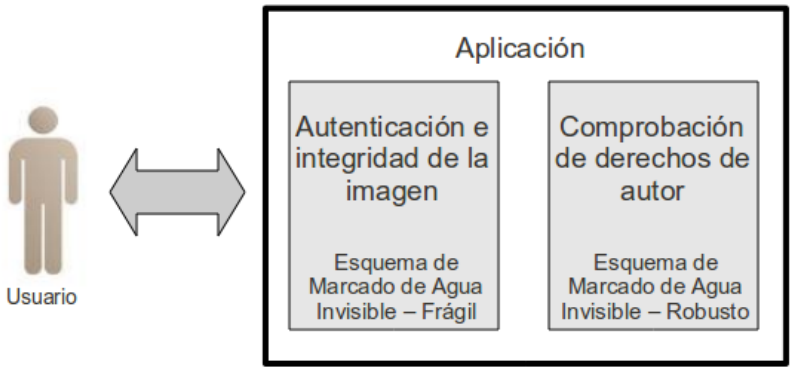



Figura 2.3: Servicios de la aplicación

La mayoría de los trabajos de investigación dedicados al estudio de las marcas de agua utilizan los conceptos descritos anteriormente para establecer un marco de trabajo que brinde servicios de seguridad. En [2] se propone un esquema de marcado de agua para dispositivos móviles que utiliza el número de celular más el código internacional de país como marca de agua y un algoritmo de inserción en el dominio de la frecuencia que utiliza la transformada discreta del coseno (DCT).

En [4] se utiliza un esquema de marcas de agua con un algoritmo de inserción en el dominio de la frecuencia que utiliza la tranformada discreta de Wavelet, sin embargo el tipo de arquitectura que se propone hace uso de un servidor proxy que es el encargado de realizar las tareas que involucran mayor procesamiento y dejan el dispositivo móvil con poca carga de trabajo.

En el ámbito comercial, existen productos disponibles al público que ponen en práctica el uso de marcas de agua, algunos de ellos son [5]: SARI (*Self-Authentication and Recovey Images*),DSS (*Digital Signature Standard*) de *Kodak*, reconocido por el NIST, IAS (*Image Authentication System*) de *Epson*, *Veridata* de *Signum Technologies*, *Eikonamark* de *AlphaTec Ltd.*, *Mediasign* de *MediaSec* y *PhotoCheck* de *AlpVision*. Los sistemas de *Kodak* y *Epson* vienen integrados en las cámaras fotográficas que producen. Otro de los sistemas de marcado de agua que actualmente existen en el mercado es distribuido por *Digimarc*, el cual brinda servicios de seguridad para la protección de derechos de autor e identificación de propiedad mediante un sistema de marcas de agua [7].

Al igual que estos sistemas, existen trabajos de investigación que se orientan a la construcción de esquemas de marcado de agua sobre dispositivos de hardware (FPGAs, GPUs e ICs). En [3] se presentan varios trabajos sobre la implementación en hardware de distintos algoritmos de marcado de agua, entre los cuales podemos encontrar: invisibles-frágiles, invisibles-robustos y visibles.

En [9] se presenta un esquema de marcado de agua de tipo espacial, el cálculo es bastante rápido ya que hace uso de operaciones matemáticas sencillas como son sumas, restas

y divisiones entre dos (que se pueden traducir en un corrimiento). El tamaño de la marca de agua debe ser igual al de la imagen a marcar y debido a que tanto la imagen como la marca de agua original son requeridas durante el proceso de detección de la marca el uso de memoria se incrementa.

Al-Gindy *et. al.* [10], presentan un esquema de marcado de agua ciego que funciona para proporcionar el servicio de protección de derechos de autor, como es un esquema que trabaja en el dominio de la frecuencia es más robusto contra diferentes ataques. Otra de sus ventajas es que permite insertar múltiples copias de la marca de agua dentro de la imagen. Debido al hecho de que utiliza la DCT, aumenta la cantidad de procesamiento requerido tanto para insertar la marca como para extraerla. Otra desventaja que trae como consecuencia el uso de la DCT es que utiliza bloques de 8x8 píxeles, para procesar la imagen, debido a esto sus medidas deben ser múltiplos de ocho o bien utilizar algún tipo de *padding*.

En [2] el esquema de marcas de agua trabaja con las imágenes en el modelo RGB por lo que no se limita a ningún tipo de formato de archivo, debido a que es un algoritmo en el dominio de la frecuencia (utiliza la DCT) y que permite insertar varias veces la marca de agua incrementa su robustez contra varios ataques. La marca de agua que se genera, es el número celular más el código internacional del país. Dentro del esquema de marcas de agua hay un proceso llamado DCS, el cual hace uso de un algoritmo de búsqueda lo que implica un costo computacional extra. Por otro lado, la función de inserción hace uso de dos funciones de cuantificación: una hacia el número par más cercano y la otra hacia el número impar más cercano.

Mohanty [13], propuso un esquema de marcado de agua que da soporte tanto para imágenes en escala de grises como para imágenes a color. Además de utilizar marcas de agua, utiliza funciones criptográficas lo cual aumenta su nivel de seguridad. Como es un esquema que utiliza un algoritmo que trabaja en el dominio de la frecuencia utiliza la DCT, esto implica mayor cantidad de procesamiento al igual que las funciones criptográficas que generalmente son más costosas en términos computacionales.

En cuanto a los esquemas de marcas de agua frágiles, una de las primeras técnicas usadas para la detección de alteraciones es el algoritmo propuesto por Walton [14] en 1995. El algoritmo consiste en seleccionar, de acuerdo a una llave secreta, grupos pseudoaleatorios de píxeles. El valor de la suma de comprobación se obtiene mediante la sumatoria de las cantidades determinadas por los siete bits más significativos del valor de los píxeles seleccionados, después los bits de la suma de comprobación son insertados en los bits menos significativos.

El proceso de verificación es similar al proceso de inserción, consiste en comparar, para cada bloque, el valor de la suma de comprobación obtenida de los valores de los píxeles de la imagen que se está verificando con los valores de la suma de comprobación recuperados de los bits menos significativos. Sin embargo, el algoritmos es susceptible cuando se intercambian bloques homólogos (bloques en la misma posición) de dos imágenes protegidas

por ls misma llave.

Li and Chang [15] propusieron un algoritmo de marcas de agua semi-frágil que acepta compresión con pérdida JPEG y rechaza ataques maliciosos. Este algoritmo trabaja en el dominio de la frecuencia, lo que significa mayor cantidad de procesamiento debido a las operaciones que deben calcularse.

Sidiropoulos *et. al.* [11], utilizaron la teoría del caos para generar la marca de agua, debido a esto el sistema es muy sensible y puede detectar fácilmente las imágenes que han sido atacadas. Otra ventajas que presenta el algoritmo es que la función de inserción es sumamente sencilla y no implica un alto costo computacional. Sin embargo, dentro de las funciones matemáticas que utiliza esta la tangente hiperbólica, la cual es más costosa computacionalmente. Aunque el uso de la teoria del caos significa grandes ventajas para el esquema, también representa ciertas desventajas ya que es suceptible al efecto de sincronización y al efecto mariposa (ver Apéndice A).

Lee and Lin [12], proponen un esquema de marcado de agua frágil e invisible, su principal ventaja es que permite insertar dos copias de la marca de agua en cada bloque de la imagen, de esta forma se proporciona una segunda oportunidad de recuperar la información de dicho bloque. Debido a esta propiedad su eficiencia en la detección de manipulaciones y recuperación de información de la imagen es muy buena. Dentro de las desventajas que presenta es que debido al tratamiento que se le da a la imagen sus medidas deben ser múltiplos de dos o utilizar algún tipo de *padding*. El proceso de detección de errores utiliza un algoritmo jerárquico de tres niveles, lo cual implica recorrer tres veces la imagen, esto a nivel computacional representa un costo extra.

| Esquema de marcado de agua | Dominio de trabajo | Tipo de marca de agua | Tipo de sistema |
|---|---|---|---|
| Mohanty *et. al.* [9] | Espacial | Robusta-invisible | No-ciego |
| Al-Gindy *et. al.* [10] | Frecuencia | Robusta-invisible | Ciego |
| Al-Gindy *et. al.* [2] | Frecuencia | Robusta-invisible | Ciego |
| Mohanty [13] | Frecuencia | Robusta-invisible | No-Ciego |
| Sidiropoulos *et. al.* [11] | Espacial | Frágil-invisible | Ciego |
| Lee & Lin [12] | Espacial | Frágil-invisible | Ciego |

Cuadro 2.1: Tabla comparativa de los diferentes esquemas de marcado de agua

En el Cuadro 2.1 se presentan las características más importantes de algunos de los esquemas de marcado de agua que se encuentran disponibles en la literatura, pero todos ellos presentan algunas similitudes en cuanto a su modo de trabajo. Algunos algoritmos utilizan como marca de agua una imagen generada de forma pseudo-aleatoria[9, 11] y con características especiales como son: el rango de valores utilizados y el tamaño de la marca, mientras que otro algoritmo utiliza como marca de agua la misma imagen [12]. Existen también algoritmos que dan más libertad en cuanto a la selección de la marca de agua [2, 10].

Generalmente los algoritmos que trabajan en el dominio de la frecuencia [2, 10, 13] hacen uso de la DCT en bloques de 8x8 píxeles y se dice que son más seguros contra cierto tipo de ataques, aunque su costo computacional sea más elevado en comparación con los algoritmos espaciales [9, 11, 12].

Por otro lado, si utilizamos como criterio de comparación el tipo de aplicación al que va orientado el esquema de marcado de agua nos encontramos que las marcas de agua robustas son utilizadas para proporcionar protección a los derechos de autor y su fin es hacer que la marca de agua no se pierda a pesar de que la imagen haya sido manipulada. En cambio las marcas de agua frágiles se utilizan para autenticar y verificar la integridad de las imágenes, una característica deseable en este tipo de esquemas de marcado de agua es que tengan soporte no solo para la detección de manipulaciones sino para la recuperación de la información debido a ellas.

El hecho de investigar y estudiar diversos esquemas de marcado de agua nos proporciona un amplio panorama de cómo es que están formados, además nos permite puntualizar las diferencias que existen en el uso de algún esquema de marcado de agua según los requerimientos que se tengan en determinado momento.

A la hora de construir una aplicación que haga uso de esquemas de marcado de agua, es indispensable tener bien definidos los alcances y limitaciones de la misma, ya que de ello dependerá el tipo de esquema que va a elegir y las características específicas que deba tener.

Para el desarrollo de este trabajo se elegieron dos esquemas de marca de agua que nos permiten cumplir con los objetivos previamente mencionados, para el servicio de autenticación de imágenes se escogió el esquema frágil descrito en [11], ya que cuenta con muchas de las características deseables para una buena implementación. A continuación se mencionan las más destacas: es invisible, ésta característica ayuda a mantener una buena calidad visual de la imagen, éste esquema de marcado trabaja en el dominio espacial por lo que el tipo de operaciones computacionales son más sencillas y para finalizar es ciego, ésta característica ayuda a las condiciones de movilidad del esquema dentro del dispositivo móvil ya que para verificar la marca de agua no es necesaria la imagen original.

Para proporcionar el servicio de protección de derechos de autor se eligió el esquema de marca de agua robusto presentado en [10], y que de acuerdo a las metas plantedas para este proyecto presenta un conjunto de cualidades adecuado, de entre las cuales podemos destacar: permite seleccionar la fuerza de inserción de la marca de agua, la cual está directamente relacionada con la calidad visual de la imagen resultante. Aunque es un esquema que trabaja en el dominio de la frecuencia, utiliza la DCT como transformada y ésta última posee muchas propiedades matemáticas que permiten hacer implementaciones eficientes.

Otra de la características importantes es que al igual que el esquema frágil, éste también es ciego. En cuanto al tipo de marca de agua que utiliza este esquema, es importante resaltar que puede ser cualquier secuencia binaria por lo que deja abierta esta opción a muchas posibilidades. En el siguiente capítulo de este documento se presenta la descripción más detallada de cada uno de estos esquemas de marcado.

# Capítulo 3

# Esquemas de marcas de agua

La finalidad de este capítulo es propocionar un panorama completo de los esquemas de marcado de agua que han sido seleccionados. En primera instancia se presenta el esquema frágil de marcado de agua, el cual será utilizado para brindar el servicio de integridad y autenticación de imágenes digitales, posteriormente se expone el esquema de marcado de agua robusto, el cual se empleará para la protección de derechos de autor. Para ambos esquemas se describe detalladamente cada una de las etapas que lo conforman con el fin de tener una mejor comprensión sobre su funcionamiento y poder determinar los alcances y limitaciones de cada uno de los esquemas.

## 3.1. Esquema frágil de marcado de agua

El esquema de marcado de agua descrito en [11] corresponde a un esquema de marcado de agua frágil, este tipo de esquemas generalmente se usan para verificar la integridad y autenticidad de imágenes digitales. Las principales características de dicho esquema son: trabaja en el dominio espacial, es ciego, es invisible, es invertible, utiliza una marca de agua que es dependiente de la información de la imagen original, utiliza la teoría del caos (ver Apéndice A)para incrementar su sensibilidad en la detección de posibles ataques y es capaz de localizar las regiones de la imagen que han sido alteradas.

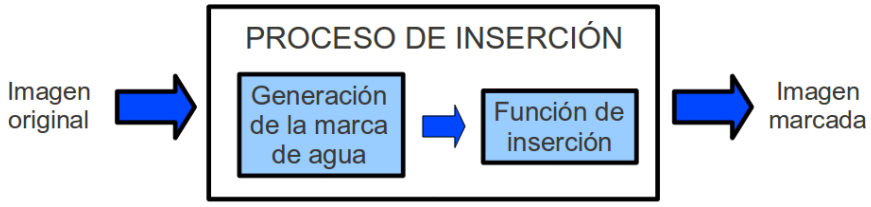



Figura 3.1: Proceso de inserción de la marca de agua

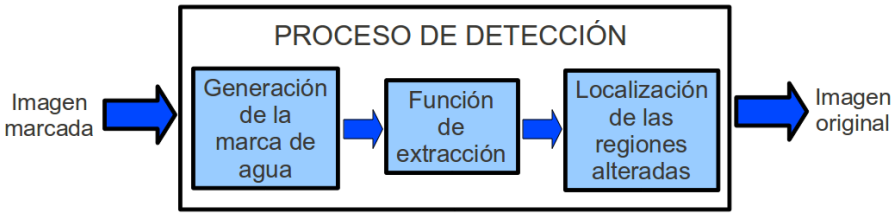



Figura 3.2: Proceso de detección de la marca de agua

En general, el esquema frágil de marcas de agua consta de dos etapas principales: la inserción y la detección de la marca de agua, figuras 3.1 y 3.2, respectivamente. A continuación se describe más detalladamente cada una de sus etapas.

### 3.1.1. Generación de la marca de agua

El proceso de generación de la marca de agua se aplica en el dominio espacial de la imagen. Consiste en recorrer por filas la imagen completa iniciando en la esquina superior izquierda mientras se evalua de forma iterativa la variable $X$, tal y como se muestra a continuación en la ecuación (3.1):

$$X_i = f(X_{i-1}) + m \cdot Y_i \tag{3.1}$$

Donde $Y_i$ es la luminancia del i-ésimo píxel, $m$ es un factor de escalamiento y $f$ es una función no lineal llamada función caótica de Chebyshev, la cual se calcula de forma iterativa de acuerdo a la ecuación (3.2):

$$f_{CHEB}(X_{i-1}) = tanh(C_1 \cdot X_{i-1}) - b \cdot tanh(C_2 \cdot X_{i-1}) \tag{3.2}$$

Para que la función no lineal $f$ tenga un comportamiento pseudo-aleatorio, es esencial que los valores que genera estén uniformemente distribuidos en el rango de $[-1, 1]$, para lograr esto, los parámetros del sistema ($b$, $C_1$ y $C_2$) deben ser elegidos de una forma adecuada. En esta implementación se tomaron como sugerencia de [11] los siguientes valores : $b = 1{,}6$, $C_1 = 200$ y $C_2 = 2$.

Después de haber calculado la secuencia pseudo-aleatoria $X$ usando la ecuación 3.1, la marca de agua binaria $W$ se genera cuantificando dicha señal con la función signo. Tal y como se muestra en la ecuación (3.3):

$$W_i = sgn(X_i) \tag{3.3}$$

Donde $sgn(X_i)$ está definida en la ecuación (3.4):

$$sgn(X_i) = \begin{cases} 1, & \text{si } X_i > 0 \\ -1, & \text{si } X_i < 0 \end{cases} \tag{3.4}$$

### 3.1.2. Inserción de la marca de agua

La marca de agua $W$ se inserta en la imagen original de forma aditiva tal y como se muestra en [11], generando así la imagen marcada $Y_W$, lo anterior representa con la ecuación (3.5):

$$Y_{W_i} = Y_i + W_i \tag{3.5}$$

### 3.1.3. Detección de la marca de agua

La detección de la marca de agua se realiza de forma ciega, es decir, sin la necesidad de recurrir a la imagen original. Comenzando por el valor $X_1$, podemos revertir el proceso de inserción de la marca basándonos en el hecho de que las ecuaciones (3.1), (3.3) y (3.5) se cumplen, debido a esto $X$ debe satisfacer una de dos desigualdades. Más concretamente, utilizando las ecuaciones (3.1) y (3.5) podemos escribir:

$$X_i = f(X_{i-1}) + m \cdot (Y_{W_i} - W_i) = f(X_{i-1}) + m \cdot Y_{W_i} - m \cdot W_i \tag{3.6}$$

Si denotamos a $A_i$ como $f(X_{i-1}) + m \cdot Y_{W_i}$, la ecuación anterior toma la siguiente forma:

$$X_i = A_i - m \cdot sgn(X_i) \tag{3.7}$$

Para cada píxel el valor y el signo de $X$ no se deben contradecir entre si, más analiticamente tenemos:

$$X_i > 0, sgn(X_i) = 1 \Rightarrow X_i = A_i - m \cdot sgn(X_i) = A_i - m > 0 \Rightarrow A_i > m$$
$$X_i < 0, sgn(X_i) = -1 \Rightarrow X_i = A_i - m \cdot sgn(X_i) = A_i + m < 0 \Rightarrow A_i < -m$$

Así pues, si los parámetros en el proceso de detección son los mismos que se utilizaron en el proceso de inserción y si la imagen no ha sido alterada el valor de $A_i$ nunca deberá estar en el intervalo de $[-m, m]$. Esta propiedad es utilizada para detectar posibles alteraciones en la imagen, pero si la imagen es auténtica dicha propiedad puede utilizarse para remover la marca de agua y recuperar la imagen original, es por esto que se dice que el algoritmo es invertible. Para llevar a cabo la detección de la marca de agua se debe hacer lo siguiente:

1. $A_i = f(X_{i-1}) + m \cdot Y_{W_i}$
2. Si $A_i > m$ entonces $W_i = 1$ y $X_i = A_i - m$
3. Si $A_i < -m$ entonces $W_i = -1$ y $X_i = A_i + m$

Si $-m < A_i < m$ se cumple para un solo píxel, entonces se puede decir que los parámetros del sistema son incorrectos o que se detectó una modificación en la imagen, el proceso se termina inmediatamente y la imagen se clasifica como no auténtica. Si el proceso de detección finaliza y ningún valor de $A_i$ estuvo dentro del intervalo $[-m, m]$, entonces la imagen se clasifica como auténtica y la imagen original puede ser recuperada restando la marca de agua obtenida en el proceso de detección a la imagen marcada.

### 3.1.4. Localización de regiones alteradas

Una de las características más deseables en un esquema frágil de marcado de agua es que sea capaz de localizar las regiones que han sido alteradas debido a un ataque. La extrema sensibilidad a modificaciones en la imagen que presenta este esquema puede ser utilizada para lograr buenos resultados en la localización de regiones alteradas.

Es importante mencionar que el valor del factor de escalamiento ($m$) que se haya elegido repercutirá en el número de píxeles que se tengan que procesar antes de detectar alguna modificación en la imagen. En [11] se reporta que cuando el valor de $m$ esta entre 1/300 y 1/400 la localización de la modificación sucederá después de escanear alrededor de 268 - 290 píxeles después de haberse encontrado el primer píxel modificado.

La precisión del proceso de localización de regiones alteradas dentro de una imagen puede mejorarse insertando varias marcas de agua en forma piramidal, es decir, en primera instancia se marca la imagen completa, posteriormente se divide en bloques no sobrepuestos de $M \times N$ píxeles y cada bloque es marcado independientemente, este proceso se repite sucesivamente hasta que el tamaño del bloque sea suficiente para un buen nivel de robustez. La principal desventaja de este modelo piramidal es que a la larga la calidad visual de la imagen se puede degradar.

Durante el proceso de detección de las marcas de agua, cada nivel es tratadado independientemente comenzando por el nivel superior, es decir, el que tiene los bloques más pequeños. Es importante mencionar que dentro de cada nivel una vez que la marca ha sido detectada se debe eliminiar, para poder pasar al siguiente.

## 3.2. Esquema robusto de marcado de agua

El esquema robusto de marcado de agua utilizado para proporcionar el servicio de protección de derechos de autor es presentado en [10], sus principales características son: trabaja en el dominio de la frecuencia, es invisible, es un esquema ciego y permite insertar multiples copias de la marca de agua en la imagen original.

Para brindar un mayor grado de robustez contra los ataques típicos, se emplea una permutación aleatoria tanto para los elementos de la marca de agua como en el orden de inserción de la misma dentro de los bloques de la imagen original, para generar las permutaciones aleatorias se utilizó un registro de desplazamiento con retroalimentación lineal (del inglés *Linear Feedback Shift Register* (LFSR), ver Apéndice B). Como se mencionó anteriormente, se pueden insertar múltiples copias de la marca de agua, el número de ellas depende directamente tanto del tamaño de la imagen original (portadora) como del tamaño de la marca de agua.

En las secciones posteriores se describe y estudia de una forma mas detallada cada una de las etapas que conforman al esquema de marcado y sus principales características.

### 3.2.1. Generación de la marca de agua

Para este esquema de marcado de agua no hay un procedimiento específico para la generación de la marca de agua ($w$), el único requerimiento es que sea binaria, es decir, cada elemento de la marca $w_i \in \{0, 1\}$.

Siendo así, para este algoritmo podemos usar casi cualquier sucesión binaria: una construida a partir de un generador de números pseudo-aleatorios, los valores binarios de una secuencia alfanmerica, los valores de una imagen digital, entre muchas otras.

### 3.2.2. Inserción de la marca de agua

El proceso de inserción de la marca de agua hace uso de la transformada discreta del coseno (DCT, ver Apéndice C), ésta se aplica a la imagen dividiendola en bloques sin translape de 8x8 píxeles. Dentro de cada uno de los bloques es posible insertar hasta ocho elementos de la marca de agua en la banda media de frecuencias. Se utiliza esta banda

con el fin de tener un equilibrio entre la robustez del esquema y la degradación visual de la imagen.

Para comenzar con el proceso de inserción es necesario aplicar un proceso de mezclado a los elementos de la marca de agua, esto con la finalidad de mejorar la robustez del esquema. Para insertar un elemento de la marca de agua se utilizan dos coeficientes horizontales adyacentes que pertenezcan a la banda media de frecuencias (ver Fig. 3.3).

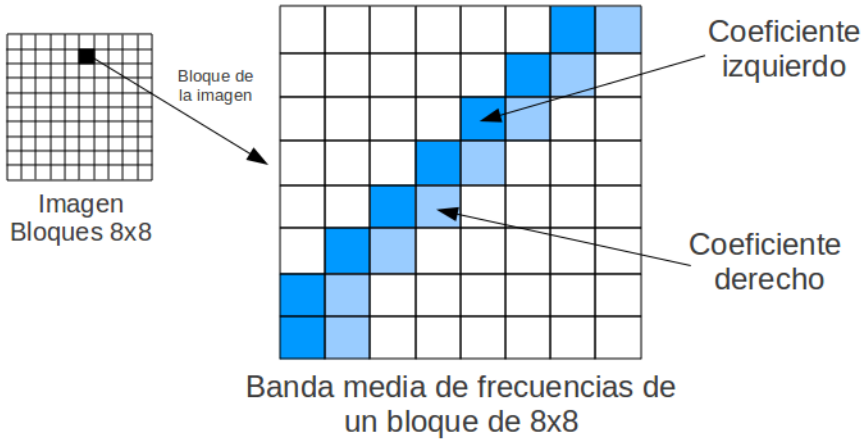



Figura 3.3: Banda media de frecuencias (coeficientes izquierdos y derechos)

La inserción de cada uno de los elementos de la marca de agua, se hace de acuerdo a la siguiente ecuación:

$$\text{Banda media } = \begin{cases} \text{Si } w_i = 1, & coef\_izq = promedio - delta \\ & coef\_der = promedio + delta \\ \text{Si } w_i = 0, & coef\_izq = promedio + delta \\ & coef\_der = promedio - delta \end{cases} \tag{3.8}$$

Donde *delta* es una constante, $w_i$ es el i-ésimo elemento de la marca de agua que se va a insertar y la variable *promedio* se calcula de la siguiente forma:

$$promedio = \frac{(coef\_izq + coef\_der)}{2} \tag{3.9}$$

La operación de inserción de un elemento se repite hasta insertar ocho elementos de la marca de agua en un bloque de la imagen, entonces se procede a calcular la DCT inversa

del mismo bloque.

Posteriormente el proceso de inserción de ocho elementos de la marca de agua se repite hasta que una marca de agua quede completamente insertada en una porción de la imagen original. Finalmente, todo el proceso de inserción de una marca de agua se repite hasta que se inserten las otras copias de la misma.

El número de copias de la marca de agua que se puede insertar dentro de la imagen original depende directamente de ambos tamaños y se puede calcular de la siguiente forma:

$$W_c = \frac{N_{HB}}{N_{WB}} \tag{3.10}$$

Donde $W_c$ es el número de copias de la marca de agua, $N_{HB}$ es el número de bloques de 8x8 píxeles que tiene la imagen original y $N_{WB}$ es número de bloques de ocho elementos que conforman a la marca de agua.

### 3.2.3. Detección de la marca de agua

El proceso de detección de la marca de agua consiste en la reconstrucción de la misma a partir de la imagen marcada, es por esto que el esquema es ciego. Una vez que la marca ha sido recuperada se compara con la original para determinar si es válida o no. Para extraer la información de la marca de agua que está dentro de la imagen marcada se divide a ésta última en bloques sin traslape de 8x8 píxeles para posteriormente aplicarle la DCT a cada uno de los mismos.

Es importante indicar cuáles bloques tienen los mismos ocho elementos de las diferentes copias de la marca de agua, ya que para cada elemento de la misma se debe calcular la sumatoria de todos los coeficientes izquierdos y derechos en donde se insertó cada uno de los elementos de la marca de agua. De las sumatorias resultantes, la marca de agua $w$ se puede reconstruir de la siguiente manera. Para cada elemento de la marca de agua:

$$w_i = \begin{cases} 1, & \text{si } \sum coef\_der > \sum coef\_izq \\ 0, & \text{si } \sum coef\_izq > \sum coef\_der \end{cases} \tag{3.11}$$

## 3.3. Medidas de desempeño para los algoritmos

En el desarrollo del siguiente capítulo, se mostrarán los resultados obtenidos de las pruebas realizadas para valorar el funcionamiento de los esquemas de marcas de agua descritos en este capítulo. Para medir el desempeño de cada uno de ellos, se emplearán parámetros

que son comunmente utilizados por muchos autores para este fin, dichos parámetros se describen a continuación:

### 3.3.1. El valor pico de la relación señal a ruido (PSNR)

El valor pico de la relación señal a ruido (PSNR), es un parámetro usualmente utilizado en procesamiento de imágenes y es una medida relativa de la calidad de la imagen. El PSNR calcula el parecido entre dos imágenes $I$ e $\tilde{I}$, de valores de intensidad comprendidos entre 0 y 255, produciendo un resultado en decibeles (dB), el cual nos da una idea del grado de similitud entre ambas imágenes.

Para definir el PSNR es indispensable el uso del error cuadrático medio (del inglés *Mean Square Error* (MSE)), que para dos imágenes $I$ e $\tilde{I}$ de tamaño $M \times N$ se calcula como:

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left[ I(i,j) - \tilde{I}(i,j) \right]^2 \tag{3.12}$$

Donde $I(i,j)$ representa un píxel de la imagen original e $\tilde{I}(i,j)$ representa un píxel de la imagen marcada.

Una vez calculado el MSE, se puede obtener el PSNR de acuerdo a la siguiente ecuación:

$$PSNR = 10 \cdot \log_{10} \frac{255^2}{MSE}(dB) \tag{3.13}$$

Un valor bajo de MSE, significa menos error en la imagen marcada con respecto a la imagen original; lo cual se traduce en un valor grande de PSNR (en decibeles), es decir, un valor grande de PSNR es bueno ya que significa que la relación señal a ruido es grande. Los valores típicos que adopta este parámetro están entre 30 y 50 dB.

### 3.3.2. La correlación normalizada (NC)

La correlación normalizada (del inglés *Normalized Correlation* (NC)), es una medición cuantitativa de similitud. En el contexto de este trabajo, la NC se utiliza para determinar el grado de similitud entre la marca de agua original ($W$) y la marca de agua extraída ($\tilde{W}$). La ecuación que define la correlación normalizada para un par de marcas de agua de tamañ $M \times N$ se presenta a continuación:

$$NC = \frac{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} W(i,j)\tilde{W}(i,j)}{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [W(i,j)]^2} \tag{3.14}$$

En algunos esquemas de marcas de agua, la marca de agua extraída puede reconocerse de forma visual. La persona encargada de hacer el reconocimiento puede comparar de forma subjetiva la marca recuperada con la original.

Sin embargo, dicha medición subjetiva es dependiente de varios factores como: experiencia de la persona, codiciones físicas de experimentación, entre otras. Debido a ésto se utiliza una métrica como la NC, la cual proporciona un juicio objetivo de la fidelidad de la marca recuperada[16].

Los valores típicos de la NC van de 0 a 1, donde 1 significa que las marcas de agua que se están comparando están altamente correlacionadas, es decir, son idénticas. Por el contrario, 0 significa una correlación nula, o que las marcas son totalmente diferentes.

# Capítulo 4

# Implementación y pruebas sobre la plataforma móvil

El uso de los dispositivos móviles se ha incrementado notablemente en los últimos años, una de las principales ventajas que poseen los dispositivos móviles es el acceso a cualquier tipo de información (imágenes, audio, texto, etc.) en cualquier momento y desde cualquier lugar. Es por esto que este trabajo esta orientado al uso de este tipo de dispositivos.

Aunque las capacidades de hardware pueden variar de un dispositivo a otro, básicamente todos los dispositivos móviles están sujetos a limitaciones en cuanto a capacidad de almacenamiento, procesamiento y uso de la batería. Es por esto que las aplicaciones móviles deben ser diseñadas para optimizar el uso de los recursos disponibles.

En este apartado se describe la implementación de los esquemas de marcado de agua previamente descritos, sobre un dispositivo móvil. Se presentan también las consideraciones que se hicieron sobre cada esquema con el fin de hacer un mejor uso de los recursos del dispositivo. Finalmente, se exponen los resultados obtenidos de las pruebas realizadas a los esquemas de marcado de agua, dichas pruebas consisten en realizar algún tipo de ataque a la imagen marcada.

## 4.1. Banco de imágenes de prueba

Para probar y verificar el correcto funcionamiento de los esquemas de marcas de agua descritos en el capítulo anterior se realizó una serie de pruebas y ataques (que van de acuerdo al esquema que se este probando) a la imagen marcada.

Para llevar a cabo el conjunto de pruebas se seleccionaron diversas imágenes en escala de grises y a color, todas ellas de distintos tamaños para tener un panorama más amplio del desempeño de los esquemas de marcado. Dichas imágenes son ampliamente conocidas y utilizadas en procesamiento de imágenes, se eligieron así con la finalidad de que los resul-

tados obtenidos puedan ser comparados con otros trabajos afines al tema de investigación. Es importante mencionar que las imágenes utilizadas en las pruebas son archivos PNG, sin embargo la implementación realizada soporta también archivos en formato JPEG.

El banco de imágenes que se utilizó es el siguiente (Fig. 4.1):

(a)Barbara

(b)Lena

(c)Pepper

(d)Baboon

Figura 4.1: Imágenes de prueba originales

Antes de continuar con la descripción de las pruebas realizadas, es relevante mencionar que las modificaciones y ataques que se realizaron a cada una de las imágenes se hicieron fuera de línea. Es decir, las imágenes se marcaron en el dispositivo móvil, posteriormente se editaron en una computadora de escritorio o laptop y se volvieron a transferir al dispositivo móvil para aplicarles el proceso de detección/extracción de la marca de agua. Para llevar a cabo la edición de las imágenes de prueba se utilizó la herramienta GIMP.

## 4.2. Esquema frágil de marcas de agua

Debido a que el esquema frágil de marcado de agua descrito en el capítulo anterior (ver sección 3.1) utiliza la teoría del caos para generar la marca de agua, es importante saber que los sistemas caóticos son extremadamente sensibles a las condiciones iniciales, esto asegura que incluso una marca de agua muy parecida a la correcta no será bien detectada en una imagen marcada.

Desafortunadamente, utilizar la teoría del caos también tiene sus desventajas ya que hace que el sistema sea vulnerable al efecto de sincronización [11] y también al efecto mariposa. El primero de ellos se puede minimizar haciendo una buena elección del parámertro $m$ y el segundo se elimina asegurando que en ambos lados del sistema las secuencias que se estén generando sean idénticas.

En este punto, es necesario hacer énfasis en los problemas que se pueden originar debido al manejo inadecuado de la precisión numérica (decimales). Si dentro de la implementación de los algoritmos no se tiene cuidado al manejar los decimales en ambos lados del sistema, se pueden generar respuestas erroneas. Para corregir este problema, basta con asegurarnos que se este utilizando el mismo número de decimales dentro de las operaciones.

Desde el punto de vista de programación el problema se controla utilizando el mismo tipo de dato en todas las operaciones, para este trabajo se están utilizando datos del tipo *float* ya que la precisión que maneja es suficiente para los cálculos requeridos. Cabe señalar que este tipo de implementaciones es posible ya que hoy en día, la mayoría de los dispositivos móviles cuentan con una unidad de punto flotante, lo cual representa una ventaja para el desarrollo de este trabajo.

### 4.2.1. Mejoras a la localización de regiones alteradas

En [11], los autores mencionan que se puede mejorar la localización de las regiones alteradas en el esquema frágil de marcado de agua, insertando más de una marca de agua en forma piramidal, sin embargo utilizar este tipo de enfoque afecta directamente la cantidad de recursos y procesamiento que se necesita para llevar a cabo el proceso de marcado y extracción de la marca, además de que la calidad visual de la imagen también se ve perjudicada.

Debido a que todo dispositivo móvil tiene ciertas restricciones en cuanto a capacidad de cómputo, se pretende que la localización de regiones alteradas siga siendo efectiva utilizando menos recursos, para ello se propone utilizar una única capa de marcas de agua con bloques de $n \times n$ píxeles en ambos procesos del esquema de marcado de agua: proceso de inserción de la marca de agua (Fig. 4.2) y proceso de detección de la marca de agua (Fig. 4.3). Con esta modificación logramos el objetivo de hacer mas eficiente el funcionamiento general del esquema de marcado de agua.

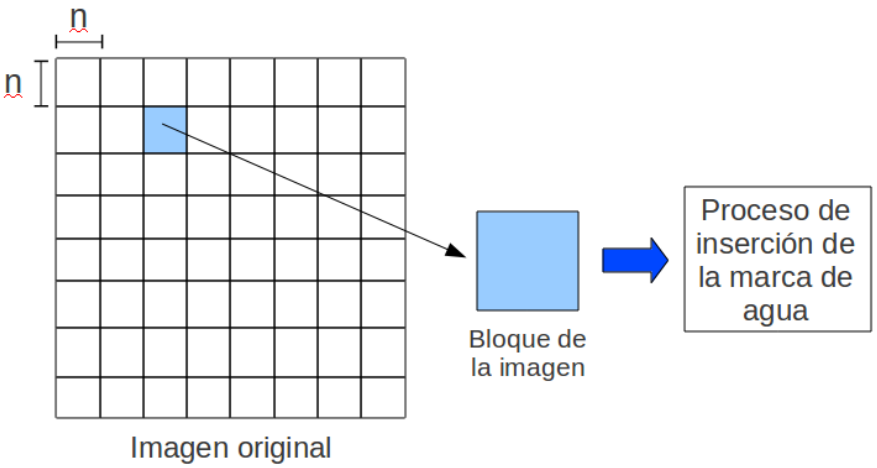



Figura 4.2: Proceso de inserción de la marca de agua

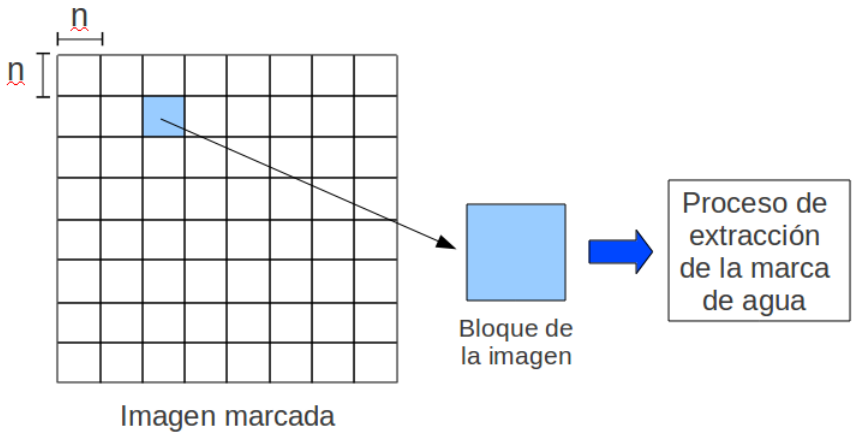



Figura 4.3: Proceso de detección de la marca de agua

El siguiente paso es elegir un tamaño de bloque adecuado, ya que éste nos indicará la granularidad de la localización de las regiones alteradas. Entre mas pequeño sea dicho bloque obtendremos una localización mas fina y exacta, por el contrario, si el tamaño del bloque es grande obtendremos un grano grueso y menos preciso.

Se hicieron diversas pruebas utilizando diferentes tamaños de bloque, con el fin de observar el comportamiento del esquema propuesto y hacer la elección correcta (ver Fig. 4.4).

(a)256x256 (b)128x128

(c)64x64 (d)32x32

Figura 4.4: Diferentes tamaños de bloque para la localización de regiones alteradas

El tamaño de bloque que finalmente se eligió es de 32x32 píxeles, ya que las imágenes utilizadas comúnmente pueden dividirse en un número suficiente de bloques de este tamaño, lo cual garantiza que la localización de regiones alteradas tenga una buena resolución, ésto se ve reflejado en el buen funcionamiento del esquema de marcado frágil.

### 4.2.2. Pruebas

De acuerdo al tamaño del bloque seleccionado para la localización de regiones alteradas, se realizarán una serie de experimentos para probar y verificar el funcionamiento del esquema frágil de marcas de agua descrito en el capítulo anterior.

Es importante recordar que la robustez no es una propiedad importante en este tipo de esquemas, sino que se detecte cualquier anomalía en la imagen marcada. Para llevar a

cabo dichas pruebas se utilizaron las imágenes que aparecen en la Fig. 4.1.

La primera prueba consiste en medir el deterioro de la imagen marcada debido al proceso de inserción, ya que lograr una buena calidad visual de la imagen despues de insertar la marca es una característica deseable en cualquier esquema de marcas de agua.

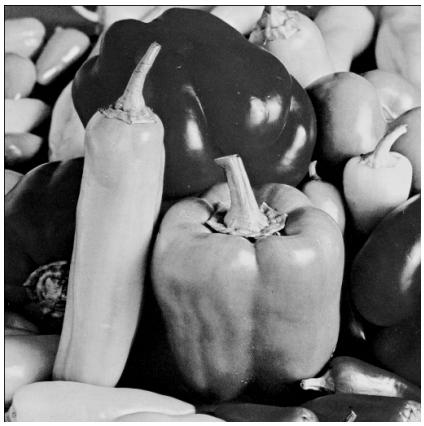
Para medir el grado de distorsión en la imagen debido al proceso de marcado se implementó un módulo para realizar el cálculo del PSNR de la imagen marcada con respecto a la versión original de la misma, a continuación se presentan tanto las imágenes marcadas como sus diferentes valores de PSNR.

PSNR = 48.1308

Para las tres

bandas (R,G,B)

Figura 4.5: Barbara después del proceso de marcado

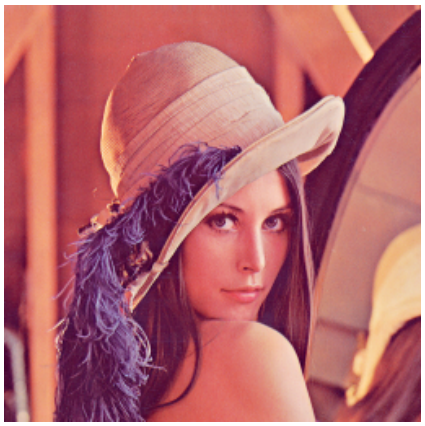


PSNR = 47.8770

Para las tres

bandas (R,G,B)

Figura 4.6: Peppers después del proceso de marcado



PSNR = 48.1317 (Banda R)

PSNR = 48.1310 (Banda G)

PSNR = 48.1308 (Banda B)

Figura 4.7: Lena después del proceso de marcado

PSNR = 48.1298 (Banda R)

PSNR = 48.1298 (Banda G)

PSNR = 48.1294 (Banda B)

Figura 4.8: Baboon después del proceso de marcado

Cabe señalar que cuando las imágenes están en escala de grises los valores de los píxeles en las tres bandas de color son exactamente iguales y como ambos esquemas de marcado insertan la marca en las tres bandas, éstas se afectan en igual medida. Debido a esto el valor de PSNR no cambia.

Para las imágenes a color no sucede lo mismo porque los valores de los píxeles son distintos en cada banda. Así que con el proceso de marcado cada banda se afecta en diferente proporción de ahí que los valores del PSNR sean distintos.

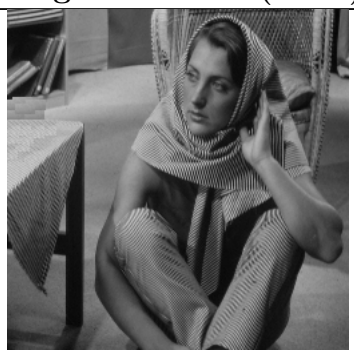
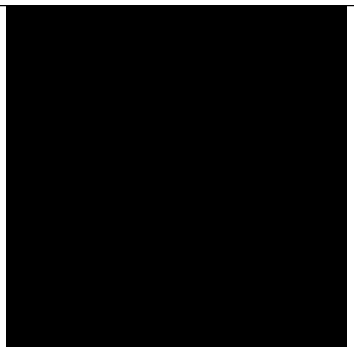
A continuación se presentan las pruebas referentes a la localización de regiones alteradas, para ello las imágenes marcadas fueron alteradas en zonas especificas, se incluyen

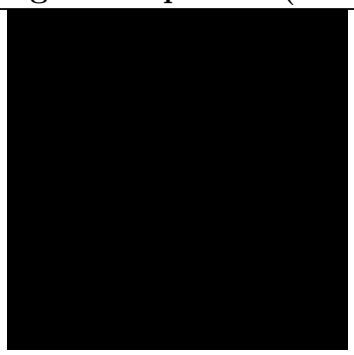
también pruebas de compresión con pérdida, ataques geométricos (giros) y adición de ruido (gaussiano y sal y pimienta).

**Barbara: 256x256 píxeles, escala de grises.**

Éstos son algunos ejemplos de los ataques que se hicieron a la imagen marcada con el fin de probar la localización de regiones alteradas dentro de la imagen marcada. En la siguiente tabla se muestran los resultados obtenidos en cada uno de los ataques realizados. La primera columna describe la modificación específica que se le hizo a la imagen y en seguida se muestra la imagen que recupera el algoritmo (Tabla 4.1).

| **Ataque** | **Imagen alterada** | **Imagen recuperada** |
|---|---|---|
| Sin ataque |  |  |
| Ataque 1: cambio de tonalidad en rostro y brazo |  |  |
| Ataque 2: cambio de tonalidad en los libros del fondo |  |  |

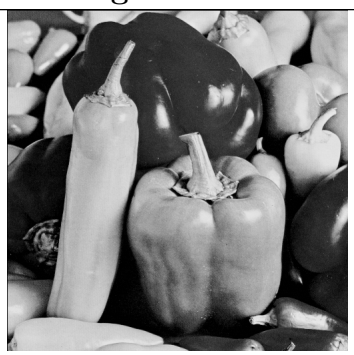
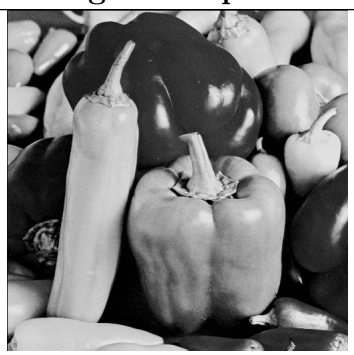
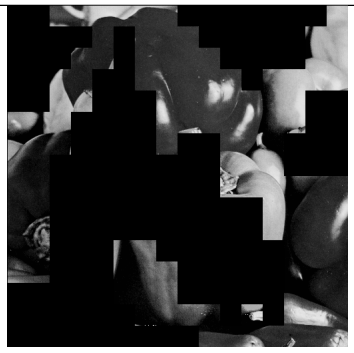
| **Ataque (cont)** | **Imagen alterada (cont.)** | **Imagen recuperada (cont.)** |
|---|---|---|
| Ataque 3: eliminación de los objetos sobre la mesa |  |  |
| Ataque 4: cambio de tonalidad en las cejas |  |  |
| Compresion JPEG (95) |  |  |
| Ruido Gaussiano ($\sigma = 15$) |  |  |

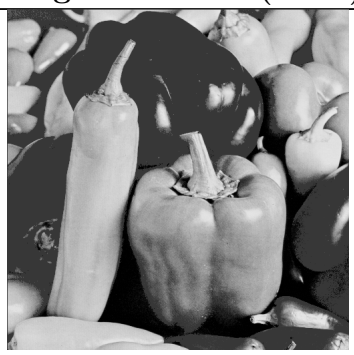
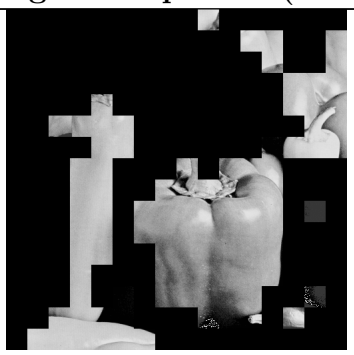
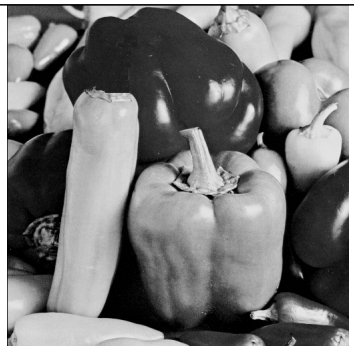
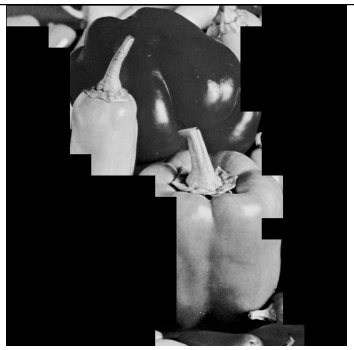
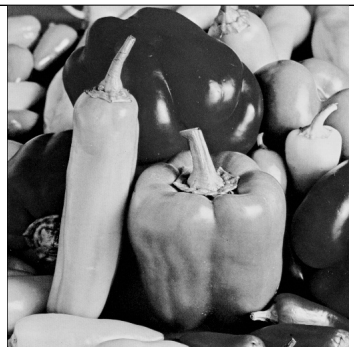
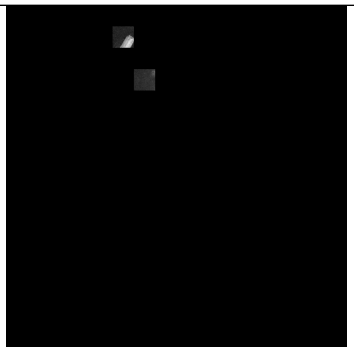
| Ataque (cont) | Imagen alterada (cont.) | Imagen recuperada (cont.) |
|---|---|---|
| Ruido S&P (Gamma = 15) |  |  |

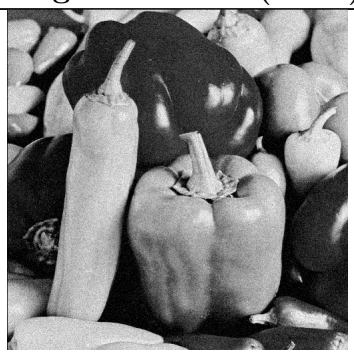
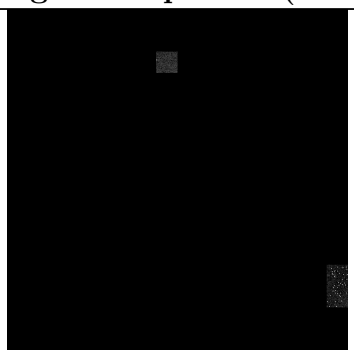
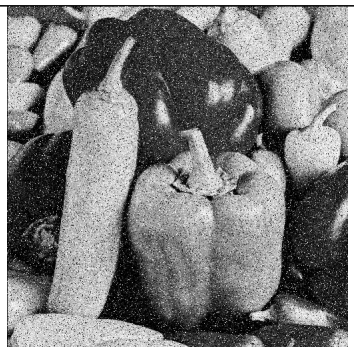
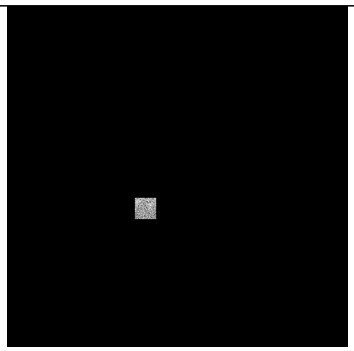
Cuadro 4.1: Barbara: Tabla de resultados de la localización de regiones alteradas.

**Peppers: 512x512 píxeles, escala de grises.**

Éstos son algunos ejemplos de los ataques que se hicieron a la imagen marcada con el fin de probar la localización de regiones alteradas dentro de la imagen marcada. En la siguiente tabla se muestran los resultados obtenidos en cada uno de los ataques realizados. La primera columna describe la modificación específica que se le hizo a la imagen y en seguida se muestra la imagen que recupera el algoritmo (Tabla 4.2).

| Ataque | Imagen alterada | Imagen recuperada |
|---|---|---|
| Sin ataque |  |  |
| Ataque 1: cambios de tonalidad en pimientos claros |  |  |

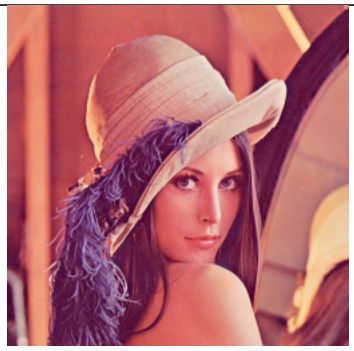
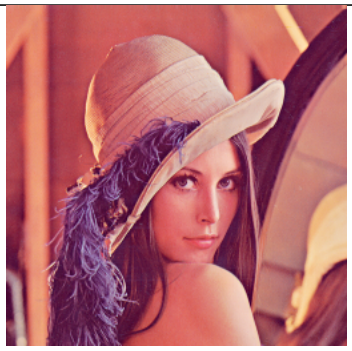
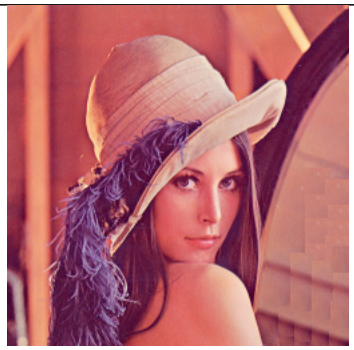
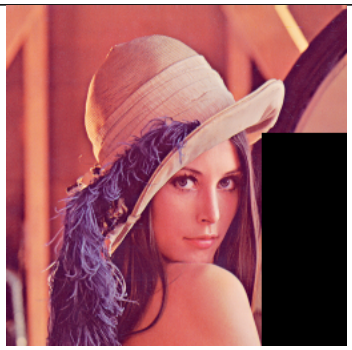
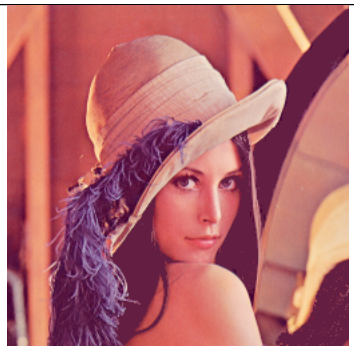
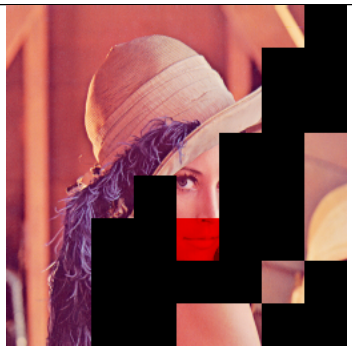
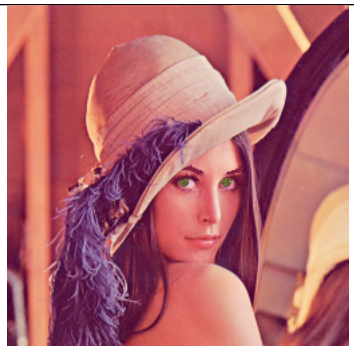
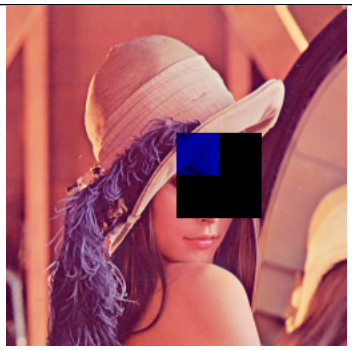
| **Ataque (cont)** | **Imagen alterada (cont.)** | **Imagen recuperada (cont.)** |
|---|---|---|
| Ataque 2: cambios de tonalidad en pimientos oscuros |  |  |
| Ataque 3: eliminación del tallo de un pimiento |  |  |
| Ataque 4: adición de pimientos a la imagen |  |  |
| Compresion JPEG (95) |  |  |

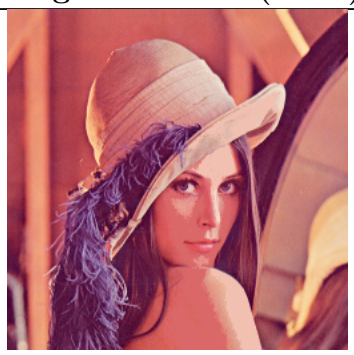
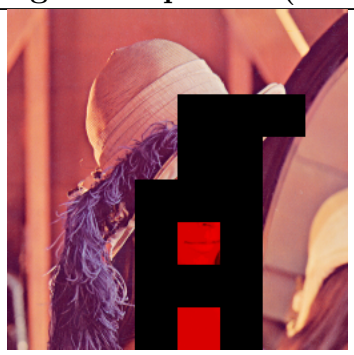
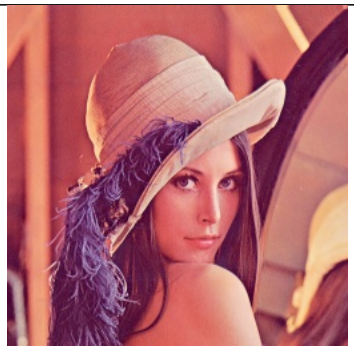
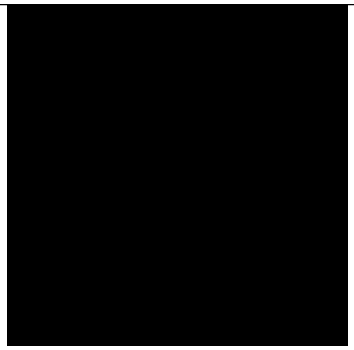
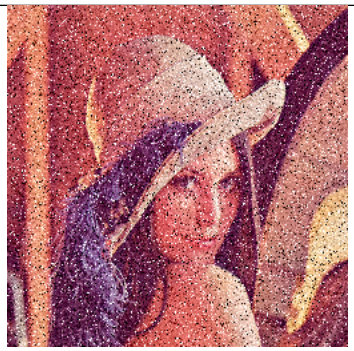
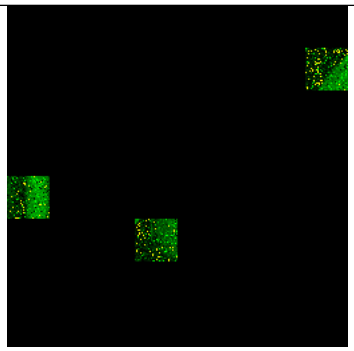
| **Ataque (cont)** | **Imagen alterada (cont.)** | **Imagen recuperada (cont.)** |
|---|---|---|
| Ruido Gaussiano ($\sigma = 15$) |  |  |
| Ruido S&P (Gamma = 15) |  |  |

Cuadro 4.2: Peppers: Tabla de resultados de la localización de regiones alteradas.

**Lena: 256x256 píxeles, a colores.**

Éstos son algunos ejemplos de los ataques que se hicieron a la imagen marcada con el fin de probar la localización de regiones alteradas dentro de la imagen marcada. En la siguiente tabla se muestran los resultados obtenidos en cada uno de los ataques realizados. La primera columna describe la modificación específica que se le hizo a la imagen y en seguida se muestra la imagen que recupera el algoritmo (Tabla 4.3).

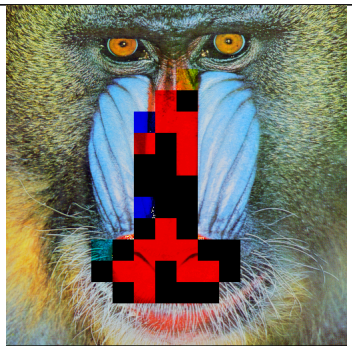
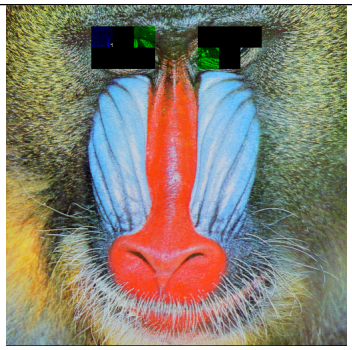
| **Ataque** | **Imagen alterada** | **Imagen recuperada** |
|---|---|---|
| Sin ataque |  |  |
| Ataque 1: eliminación del reflejo |  |  |
| Ataque 2: cambio de tonalidad en el cabello y marco del espejo |  |  |
| Ataque 3: cambio de tonalidad en los ojos y cejas |  |  |

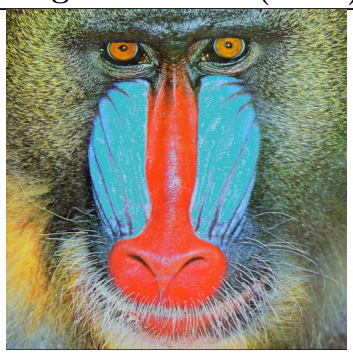
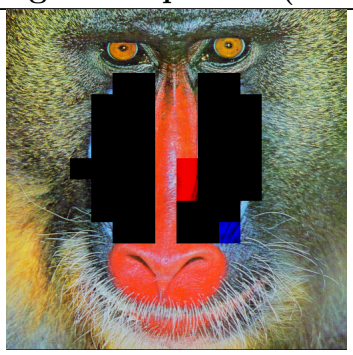
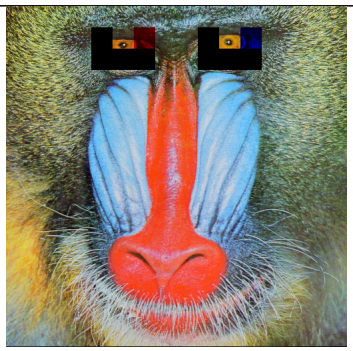
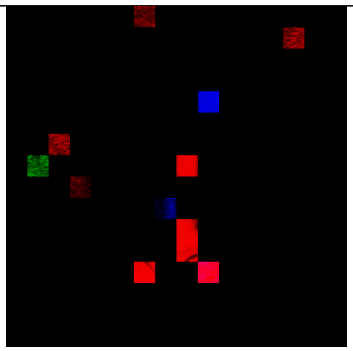
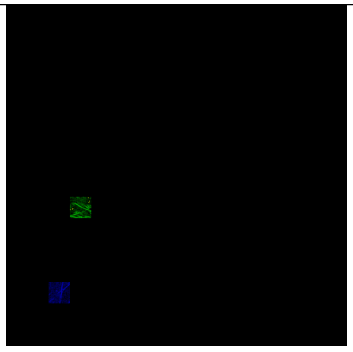
| Ataque (cont) | Imagen alterada (cont.) | Imagen recuperada (cont.) |
|---|---|---|
| Ataque 4: cambio de tonalidad en la piel |  |  |
| Compresion JPEG (95) |  |  |
| Ruido Gaussiano ($\sigma = 15$) |  |  |
| Ruido S&P (Gamma = 15) |  |  |

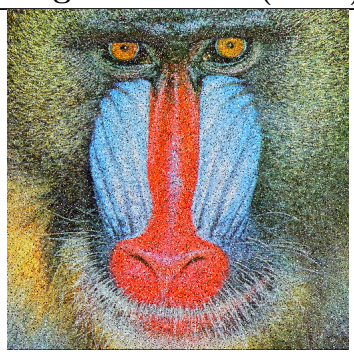
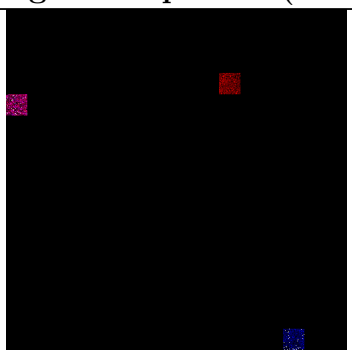
Cuadro 4.3: Lena: Tabla de resultados de la localización de regiones alteradas.

**Baboon: 512x512 píxeles, a colores.**

Éstos son algunos ejemplos de los ataques que se hicieron a la imagen marcada con el fin de probar la localización de regiones alteradas dentro de la imagen marcada. En la siguiente tabla se muestran los resultados obtenidos en cada uno de los ataques realizados. La primera columna describe la modificación específica que se le hizo a la imagen y en seguida se muestra la imagen que recupera el algoritmo (Tabla 4.4).

| **Ataque** | **Imagen alterada** | **Imagen recuperada** |
|---|---|---|
| Sin ataque |  |  |
| Ataque 1: cambio de tonalidad de la nariz |  |  |
| Ataque 2: cambio de tonalidad en los ojos |  |  |

| Ataque (cont) | Imagen alterada (cont.) | Imagen recuperada (cont.) |
|---|---|---|
| Ataque 3: cambio de tonalidad en la sección azul |  |  |
| Ataque 4: cambio de tonalidad en el contorno de los ojos |  |  |
| Compresion JPEG (95) |  |  |
| Ruido Gaussiano ($\sigma = 15$) |  |  |

| **Ataque (cont)** | **Imagen alterada (cont.)** | **Imagen recuperada (cont.)** |
|---|---|---|
| Ruido S&P (Gamma = 15) |  |  |

Cuadro 4.4: Baboon: Tabla de resultados de la localización de regiones alteradas.

### 4.2.3. Análisis de resultados del esquema frágil

En este apartado se presentan algunas reflexiones que esclarecen los resultados obtenidos de las distintas pruebas que se le hicieron al esquema frágil de marca de agua. Antes de continuar, es importante recordar que los ataques para los esquemas frágiles difieren significativamente de los ataques para los esquemas robustos. En los esquemas frágiles un atacante no esta interesado en hacer la marca de agua indectable, de hecho, destruirla es bastante sencillo debido a su naturaleza frágil.

Los ataques a un esquema frágil tienen como objetivo principal extraer información de la marca de agua para así autenticar otras imágenes, sustituir una marca de agua existente con otra que sea falsa o simplemente modificar la imagen de tal forma que dicha modificación no sea detectada. Hay gran cantidad de niveles de ataques, éstos dependen de la información y herramientas que tenga disponible el atacante.

La robustez del esquema de marcado de agua descrito anteriormente se deriva del hecho de que los valores de la marca de agua están determinados por al menos tres factores: la imagen original, los parámetros de la función caótica y el valor inicial, cualquier cambio en alguno de éstos parámetros producirá una respuesta diferente en el sistema [11].

En esta implementación se utilizaron los siguientes valores : $b = 1{,}6$, $C_1 = 200$, $C_2 = 2$ y $m = \frac{1}{350}$, los cuales son propuestos por los autores del esquema en [11].

Otro factor que debemos tener en cuenta para este análisis de resultados, es que se sustituyó el modelo piramidal de marcas de agua por una única capa de ellas, esto reduce la eficacia de la localización de regiones alteradas ya que puede suceder que una alteración no sea detectada a lo largo del proceso.

Además de lo anterior, también es importante destacar que el proceso de marcado se aplica a cada una de las bandas de color, siguiendo el modelo RGB. Cuando un bloque de la imagen es catalogado como "alterado", los valores de los píxeles correspondientes a dicho bloque se reemplazan por el valor cero. Debido a esto el color utilizado para señalar las regiones alteradas dentro de una imagen puede variar, dichas variaciones dependen esencialmente de la banda de color que se haya alterado.

| Caso | Combinación | | | Color |
|---|---|---|---|---|
| | R | G | B | |
| La alteración se detectó en las tres bandas (R,G,B) | 0 | 0 | 0 | Negro |
| La alteración se detectó en las bandas R y G | 0 | 0 | - | Tonalidades de azul |
| La alteración se detectó en las bandas R y B | 0 | - | 0 | Tonalidades de verde |
| La alteración se detectó en las bandas G y B | - | 0 | 0 | Tonalidades de rojo |
| La alteración se detectó solo en la banda R | 0 | - | - | Tonalidades de cyan |
| La alteración se detectó solo en la banda G | - | 0 | - | Tonalidades de magenta |
| La alteración se detectó solo en la banda B | - | - | 0 | Tonalidades de amarillo |

Cuadro 4.5: Posibles colores en la localización de regiones alteradas

En la tabla 4.5 se muestran las posibles combinaciones según la banda que se haya detectado como "alterada", el color puede tener ciertas variaciones dependiendo el valor de los pixeles que estén presentes en el bloque.

Para finalizar, es fundamental mencionar que este tipo de resultados son exclusivos en las imagenes a color. Para las imágenes en escala de grises solo es posible indicar las regiones alteradas con el color negro, esto es debido a que los valores de los píxeles son los mismos para las tres bandas de color.

## 4.3. Esquema robusto de marcado de agua

Generalmente los esquemas de marcado de agua robusto trabajan en el dominio de la frecuencia, de esta forma se incrementa la robustez del esquema. Para el desarrollo de este trabajo se eligió un algoritmo de marcado de agua que utiliza la DCT, dicha transformada requiere de una serie de operaciones matemáticas computacionalmente costosas. Debido a lo anterior se utilizó el cálculo rápido de la DCT, la cual se describe a continuación.

### 4.3.1. Cálculo rápido de la DCT

En esta sección solo se consideran los detalles para el cálculo rápido de la DCT, para mas detalles acerca de su definición y propiedades veáse el Apéndice C.

Es interesante mencionar que cuando deseamos aplicar la DCT a una imagen, debemos utilizar la DCT de dos dimensiones (2D-DCT). Debido a la propiedad de separabilidad (véase C.3.3), dicha transformada puede calcularse aplicando dos veces la DCT de una dimensión (1D-DCT): primero por columnas y luego por filas. Esta idea se ilustra gráficamente en la figura (Fig. 4.9). Se puede seguir el mismo modelo para calcular la transformada inversa (ecuación C.5).

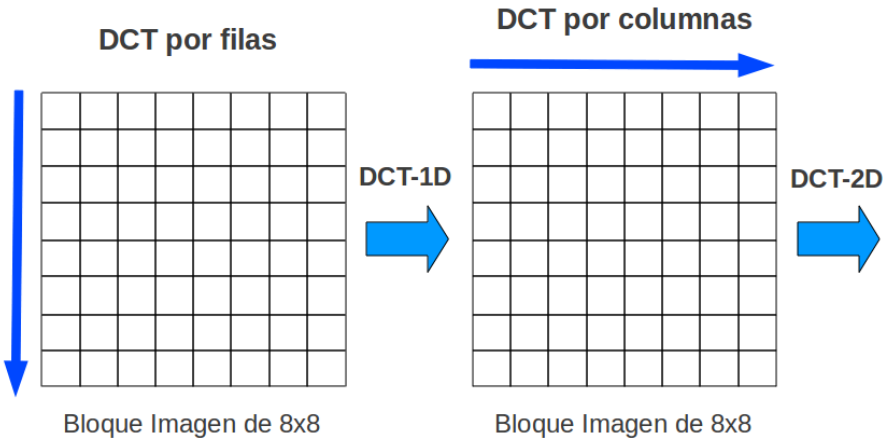
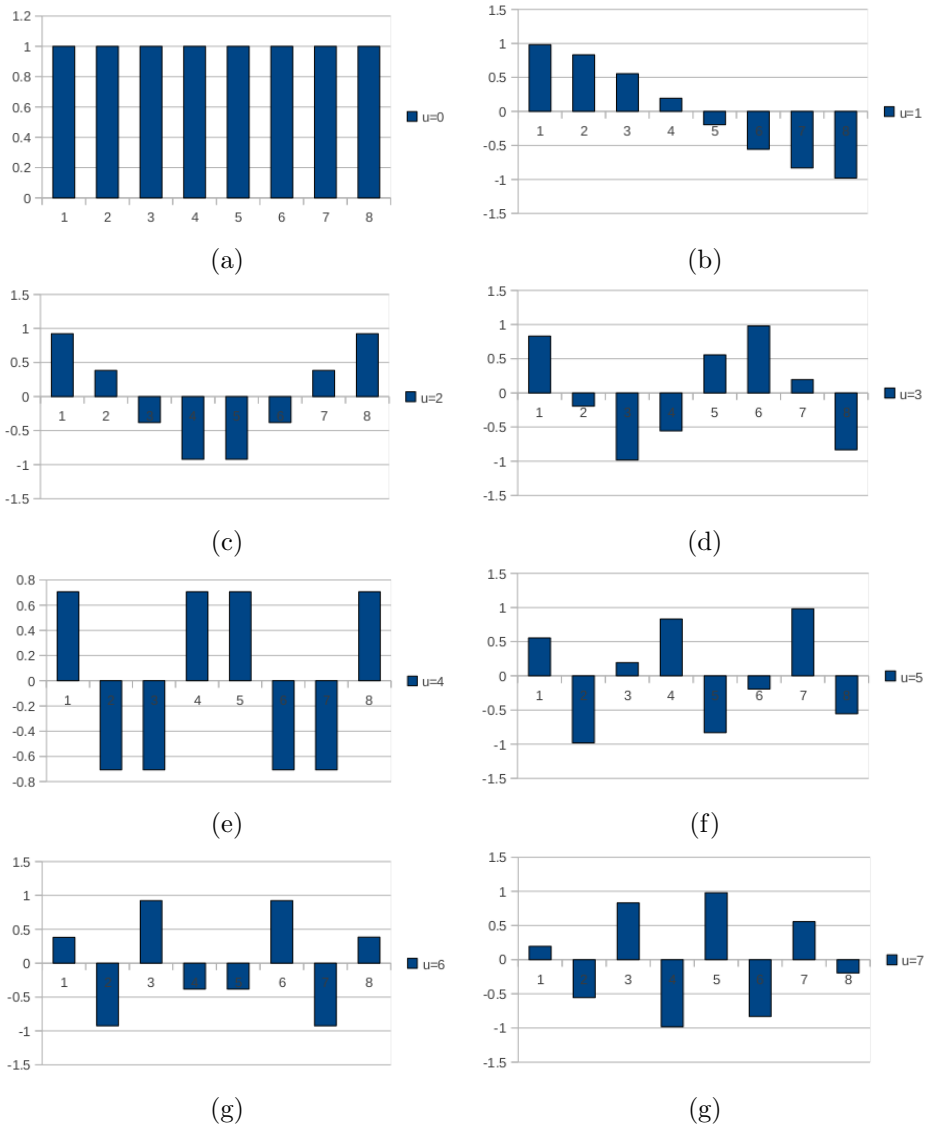



Figura 4.9: Calculo de la DCT-2D en dos pasos

Debido a lo anterior, es importante que la DCT unidimensional pueda calcularse de manera eficiente, para reducir el tiempo de cálculo se optó por utilizar tablas de consulta. A continuación se explica el procedimiento que se siguió para construir dichas tablas.

Si por un momento pensamos únicamente en la DCT unidimensional y en la ecuación que la define (ec. C.1) ignoramos los terminos $f(x)$ y $\alpha(u)$, las gráficas de $\sum_{x=0}^{N-1} cos\left[\frac{\pi(2x+1)u}{2N}\right]$ para $N = 8$ y $u = 1, 2, \ldots, 7$ presentadas en la Fig. (Fig. 4.10) nos muestran unas formas de onda que representan las *funciones bases del coseno*, las cuales son ortogonales e independientes.




(a) (b) (c) (d) (e) (f) (g) (g)

Figura 4.10: Funciones base del coseno

Es importante hacer notar que las *funciones base del coseno* no cambian para cada cálculo de la DCT, los unicos valores que varian son los de la función $f(x)$. Ésta es un propiedad muy importante, ya que se muestra que dichas *funciones base* pueden ser precalculadas y utilizadas como una tabla de consulta (ver Tabla 4.7) a la hora de calcular la 1D-DCT.

De igual forma, se puede construir una tabla de consulta (ver Tabla 4.8) para la función $\alpha(u)$ ya que solo esta definida para dos casos: cuando $u = 0$ y cuando $u \neq 0$.

Para la implementación de este trabajo se esta trabajando con bloques de 8x8 píxeles para el calculo de la 2D-DCT, así que las tablas de consulta para la 1D-DCT quedan de la siguiente manera:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0.980 | 0.831 | 0.555 | 0.195 | -0.195 | -0.555 | -0.831 | -0.980 |
| 0.923 | 0.382 | -0.382 | -0.923 | -0.923 | -0.382 | 0.382 | 0.923 |
| 0.831 | -0.195 | -0.980 | -0.555 | 0.555 | 0.980 | 0.195 | -0.831 |
| 0.707 | -0.707 | -0.707 | 0.707 | 0.707 | -0.707 | -0.707 | 0.707 |
| 0.555 | -0.980 | 0.195 | 0.831 | -0.831 | -0.195 | 0.980 | -0.555 |
| 0.382 | -0.923 | 0.923 | -0.382 | -0.382 | 0.923 | -0.923 | 0.382 |
| 0.195 | -0.555 | 0.831 | -0.980 | 0.980 | -0.831 | 0.555 | -0.195 |

Cuadro 4.7: Tabla de consulta de las *funciones base del coseno*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0.353 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 |

Cuadro 4.8: Tabla de consulta de la función $\alpha(u)$

### 4.3.2. El esquema de mezclado

Una vez que la imagen original ha sido dividida en bloques de 8x8 píxeles comienza el proceso de inserción, tipicamente dicho proceso inicia en la esquina superior izquierda de la imagen (bloque 0,0) y la va recorriendo por filas hasta marcarla por completo.

Sin embargo, este enfoque tiene una debilidad contra los cortes verticales ya que la marca de agua esta distribuida de forma horizontal dentro de la imagen. Para brindar un mayor grado de robustez contra los ataques de recorte, se emplean dos permutaciones pseudo-aleatorias: una de ellas define el orden de inserción de la marca de agua dentro de los bloques de la imagen y la otra sirve para mezclar los elementos de la marca de agua.

Para obtener una secuencia de números que funcione como una permutación pseudo-aleatoria, se utilizó un registro de desplazamiento con retroalimentación lineal (LFSR), para más detalles ver Apéndice B. Existen varias formas de construir un LFSR, los hay a nivel de hardware pero también se pueden implementar en software. En este trabajo, se desarrollo un LFSR utilizando operaciones a nivel de bits: máscaras, operaciones XOR, AND y corrimientos. Cabe señalar que los dispositivos móviles soportan este tipo de operaciones.

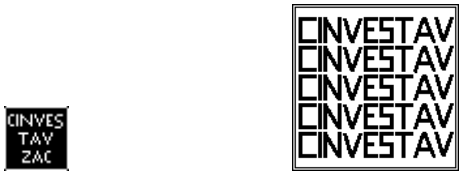
El problema al que nos enfrentamos es que un LFSR tiene asociada cierta longitud, la cual esta ligada al número de elementos que puede generar. Cubrir absolulamente todos los diferentes tamaños posibles, es casi imposible, así que solo se consideraron algunos

LFSR significativos para el desarrollo de este trabajo.

Es importante mencionar que la permutación aleatoria que se genere debe ser igual tanto en el proceso de inserción de la marca como en el proceso de extracción, esto con el fin de garantizar que la marca sea extraída de forma correcta. El valor de inicialización del LFSR puede ser utilizado como llave del sistema, de hecho es recomendable que para cada imagen se utilicen distintas semillas.

### 4.3.3. Pruebas

Como se mencionó anteriormente, para este esquema de marcas de agua podemos utilizar como marca cualquier sucesión binaria, en este caso se opto por utilizar un logo en blanco y negro, con el fin de que el reconocimiento de la marca recuperada sea tanto visual como por medio del umbral. A continuación se muestran los logos que se utilizaron:



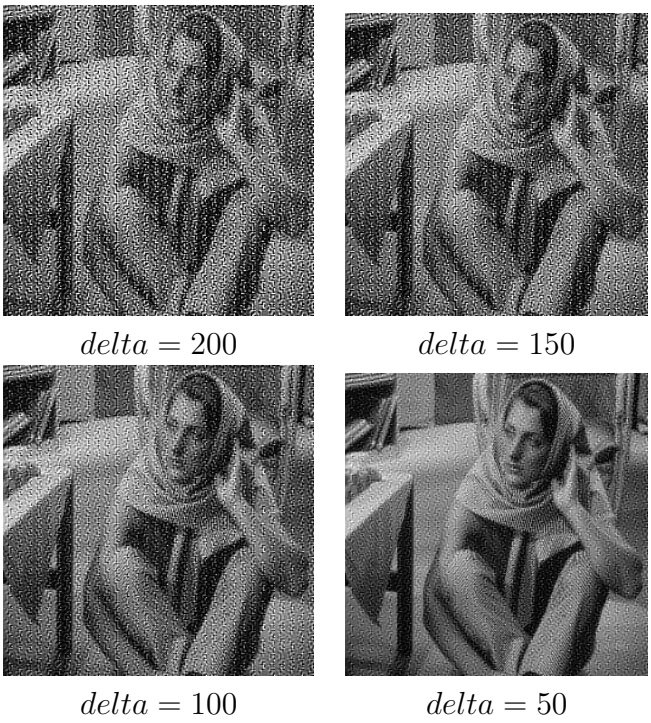

(a)Logo 1 - 32x32 píxeles (b)Logo 2 - 128x128 píxeles

Figura 4.11: Logos utilizados como marca de agua

Primero, se presentan las puebas realizadas para la elección del parámetro *delta*, el cual nos indica la fuerza de la marca de agua dentro de la imagen. Existe una relación directa entre el valor de *delta* y la degradación visual de la imagen: un valor grande del parámetro *delta* significa mayor degradación visual de la imagen.

Para hacer la elección del parámetro *delta* de forma adecuada se probaron distintos valores. En la Fig. 4.12 se muestran valores de *delta* que afectan severamente la calidad visual de la imagen. Por otro lado, la Fig. 4.13 se muestran valores de *delta* que afectan en menor medida la calidad visual de la imagen.

Finalmente, se opto por elegir $delta = 3$, ya que la degradación visual de la imagen es mínima y tiene buen grado de robustez, lo cual se constatará en las siguientes pruebas. No obstante, es posible elegir otro valor de *delta* sin olvidar que la calidad visual de la imagen disminuirá.



$delta = 200$ $delta = 150$

$delta = 100$ $delta = 50$

Figura 4.12: Valores de *delta* que degradan la calidad visual de la imagen.

$delta = 10$ $delta = 5$

$delta = 4$ $delta = 3$

Figura 4.13: Valores de *delta* que conservan la calidad visual de la imagen.

Ahora bien, entrando de lleno en el análisis de desempeño del algoritmo la primera variable que vamos a evaluar es el deterioro de la imagen marcada después del proceso de inserción, ya que mantener una buena calidad visual de la imagen despues de insertar la marca es una característica deseable en cualquier esquema de marcas de agua. Recordemos que las imágenes de prueba que se están utilizando son las que aparecen en la Fig. 4.1.

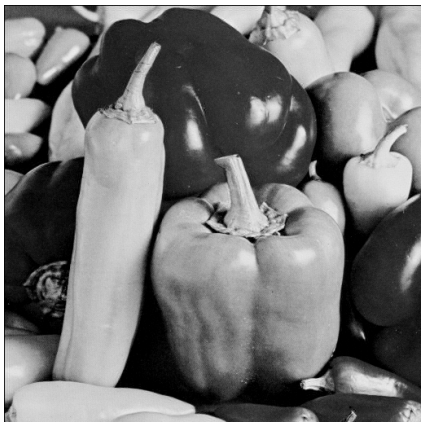
Para medir el grado de distorción en la imagen debido al proceso de marcado se calculó el PSNR de la imagen marcada con respecto a la versión original de la misma, a continuación se presentan tanto las imágenes marcadas como sus diferentes valores de PSNR.

PSNR = 34.3136

Para las tres

bandas (R,G,B)

Figura 4.14: Barbara después del proceso de marcado, utilizando el Logo 1



PSNR = 38.1514

Para las tres

bandas (R,G,B)

Figura 4.15: Peppers después del proceso de marcado, utilizando el Logo 2

PSNR = 37.4158 (Banda R)

PSNR = 35.3649 (Banda G)

PSNR = 36.7336 (Banda B)

Figura 4.16: Lena después del proceso de marcado, utilizando el Logo 1

PSNR = 30.9729 (Banda R)

PSNR = 29.5616 (Banda G)

PSNR = 29.7550 (Banda B)

Figura 4.17: Baboon después del proceso de marcado, utilizando el Logo 1

En este punto es importante resaltar que los valores de PSNR obtenidos para este esquema de marcado son menores que en el esquema frágil. Ésto se debe a que el proceso de inserción del esquema robusto es más severo ya que trabaja en el dominio de la frecuencia y el valor de los píxeles de la imagen se incrementa o disminuye en más de una unidad, lo que no ocurre en el esquema frágil.

El siguiente aspecto que vamos a valorar es la robutez del esquema de marcas de agua ante diferentes tipos de ataques, como son: ataques geométricos que incluyen cortes horizontales, verticales o una combinación de ambos; ataques de interferencia los cuales insertan cierto nivel de ruido a la imagen; ataques al esquema en si, es decir utilizar

parámetros de inserción diferentes a los originales y por último diferentes niveles de compresión con pérdida (JPEG).

Para medir el desempeño del algoritmo en cuanto a la recuperación de la marca de agua, se utilizará la correlación cruzada (NC). A continuación se muestran los valores obtenidos de NC para cada uno de los logos recuperados en cada una de las imágenes de prueba que se mencionaron al principio de la sección, además se muestran algunas de las imágenes que han sido atacadas. Cabe sealar que el proceso de extracción se aplica a cada banda de color (R,G,B).

**Lena: 256x256 píxeles, a colores. Marca de agua: Logo 1**

Éstos son algunos ejemplos de los ataques que se hicieron a la imagen marcada con el fin de probar la robustez del esquema de marcas de agua.

(a)Recorte vertical 50 % (b)Giro 90° (c)Compresión JPEG 75

(d)Corte Horizontal-Vertical 25 % (e)Ruido Gaussiano($\sigma = 20$) (f)Ruido S&P (Gamma = 0.5)

Figura 4.18: Diferentes ataques a la imagen marcada

En seguida se muestran los resultados obtenidos en cada uno de los ataques realizados. La primera tabla abarca los ataques de corte y rotación (Tabla 4.11), mientras que la segunda contiene los ataques de compresión y adición de ruido (Tabla 4.12).

| Ataque | NC | | | Logos Recuperados | | |
|---|---|---|---|---|---|---|
| | **Banda r** | **Banda g** | **Banda b** | **Banda r** | **Banda g** | **Banda b** |
| Sin ataque | 1.0 | 1.0 | 1.0 |  |  |  |
| Corte 25 % horizontal | 1.0 | 1.0 | 1.0 |  |  |  |
| Corte 25 % vertical | 1.0 | 1.0 | 1.0 |  |  |  |
| Corte 50 % horizontal | 0.9918 | 0.9918 | 0.9918 |  |  |  |
| Corte 50 % vertical | 0.9918 | 0.9918 | 0.9918 |  |  |  |
| Corte horizontal-vertical 25 % | 1.0 | 1.0 | 1.0 |  |  |  |
| Llaves incorrecta | 0.2040 | 0.2040 | 0.2040 |  |  |  |
| Giro 90°derecha | 0.4693 | 0.4326 | 0.4244 |  |  |  |
| Giro 90 °izquierda | 0.4408 | 0.4489 | 0.4530 |  |  |  |
| Giro 180° | 0.5469 | 0.5020 | 0.4938 |  |  |  |

Cuadro 4.11: Lena: Ataques de corte y rotación.

| Ataque | NC | | | Logos Recuperados | | |
|---|---|---|---|---|---|---|
| | **Banda r** | **Banda g** | **Banda b** | **Banda r** | **Banda g** | **Banda b** |
| Compresion JPEG (80) | 0.7387 | 0.7224 | 0.6612 |  |  |  |
| Compresion JPEG (85) | 0.8122 | 0.8244 | 0.8 |  |  |  |
| Compresion JPEG (90) | 0.9224 | 0.9346 | 0.9142 |  |  |  |
| Ruido Gaussiano ($\sigma = 5$) | 0.9959 | 0.9959 | 0.9959 |  |  |  |
| Ruido Gaussiano ($\sigma = 10$) | 0.8448 | 0.8408 | 0.8448 |  |  |  |
| Ruido Gaussiano ($\sigma = 15$) | 0.7714 | 0.7673 | 0.7755 |  |  |  |
| Ruido S&P (Gamma = 0.5) | 0.8857 | 0.8938 | 0.8938 |  |  |  |
| Ruido S&P (Gamma = 2.5) | 0.7020 | 0.7224 | 0.7142 |  |  |  |
| Ruido S&P (Gamma = 5) | 0.6612 | 0.6653 | 0.6653 |  |  |  |

Cuadro 4.12: Lena: Ataques de compresión y adición de ruido.

**Barbara: 256x256 píxeles, escala de grises. Marca de agua: Logo 1**

Éstos son algunos ejemplos de los ataques que se hicieron a la imagen marcada con el fin de probar la robustez del esquema de marcas de agua.

(a)Recorte horizontal 25 % (b)Recorte vertical 50 % (c)Giro 90°

(d)Compresión JPEG 75 (e)Ruido Gaussiano($\sigma = 25$) (f)Ruido S&P (Gamma = 2.5)

Figura 4.19: Diferentes ataques a la imagen marcada

En seguida se muestran los resultados obtenidos en cada uno de los ataques realizados. La primera tabla abarca los ataques de corte y rotación (Tabla 4.13), mientras que la segunda contiene los ataques de compresión y adición de ruido (Tabla 4.14).

| **Ataque** | **NC** | | | **Logos Recuperados** | | |
|---|---|---|---|---|---|---|
| | **Banda r** | **Banda g** | **Banda b** | **Banda r** | **Banda g** | **Banda b** |
| Sin ataque | 1.0 | 1.0 | 1.0 |  |  |  |
| Corte 25 % horizontal | 1.0 | 1.0 | 1.0 |  |  |  |

| Ataque (cont.) | NC (cont.) | | | Logos Recuperados (cont.) | | |
|---|---|---|---|---|---|---|
| | Banda r | Banda g | Banda b | Banda r | Banda g | Banda b |
| Corte 25 % vertical | 1.0 | 1.0 | 1.0 |  |  |  |
| Corte 50 % horizontal | 1.0 | 1.0 | 1.0 |  |  |  |
| Corte 50 % vertical | 1.0 | 1.0 | 1.0 |  |  |  |
| Corte contorno | 1.0 | 1.0 | 1.0 |  |  |  |
| Llaves incorrecta | 0.2367 | 0.2367 | 0.2367 |  |  |  |
| Giro 90°derecha | 0.4693 | 0.4693 | 0.4693 |  |  |  |
| Giro 90 °izquierda | 0.4693 | 0.4693 | 0.4693 |  |  |  |
| Giro 180° | 0.4816 | 0.4816 | 0.4816 |  |  |  |

Cuadro 4.13: Barbara: Ataques de corte y rotación.

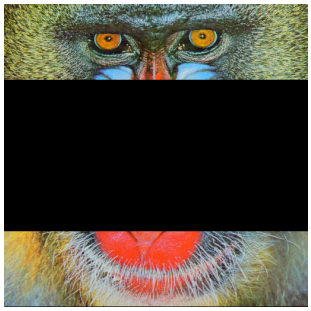
| Ataque | NC | | | Logos Recuperados | | |
|---|---|---|---|---|---|---|
| | Banda r | Banda g | Banda b | Banda r | Banda g | Banda b |
| Compresion JPEG (80) | 0.7836 | 0.7836 | 0.7836 |  |  |  |

| **Ataque (cont.)** | **NC (cont.)** | | | **Logos Recuperados (cont.)** | | |
|---|---|---|---|---|---|---|
| | **Banda r** | **Banda g** | **Banda b** | **Banda r** | **Banda g** | **Banda b** |
| Compresion JPEG (85) | 0.8857 | 0.8857 | 0.8857 |  |  |  |
| Compresion JPEG (90) | 0.9551 | 0.9551 | 0.9551 |  |  |  |
| Ruido Gaussiano ($\sigma = 5$) | 0.9877 | 0.9877 | 0.9877 |  |  |  |
| Ruido Gaussiano ($\sigma = 10$) | 0.8857 | 0.8857 | 0.8857 |  |  |  |
| Ruido Gaussiano ($\sigma = 15$) | 0.7387 | 0.7387 | 0.7387 |  |  |  |
| Ruido S&P (Gamma = 0.5) | 0.8857 | 0.8857 | 0.8857 |  |  |  |
| Ruido S&P (Gamma = 2.5) | 0.7387 | 0.7387 | 0.7387 |  |  |  |
| Ruido S&P (Gamma = 5) | 0.6653 | 0.6653 | 0.6653 |  |  |  |

Cuadro 4.14: Barbara: Ataques de compresión y adición de ruido.

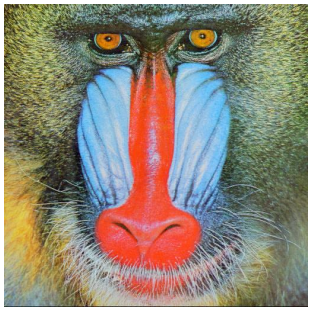
**Baboon: 512x512 píxeles, a colores. Marca de agua: Logo 1**

Éstos son algunos ejemplos de los ataques que se hicieron a la imagen marcada con el fin de probar la robustez del esquema de marcas de agua.



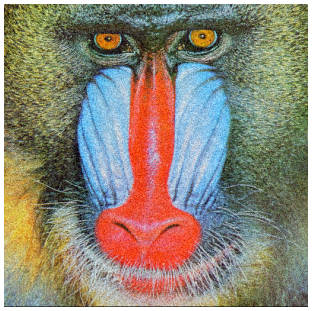
(a)Recorte horizontal 50 %

(b)Giro 180°



(c)Compresión JPEG 75

(d)Corte contorno



(e)Ruido Gaussiano($\sigma = 25$)

(f)Ruido S&P (Gamma = 5)

Figura 4.20: Diferentes ataques a la imagen marcada

En seguida se muestran los resultados obtenidos en cada uno de los ataques realizados. La primera tabla abarca los ataques de corte y rotación (Tabla 4.15), mientras que la segunda contiene los ataques de compresión y adición de ruido (Tabla 4.16).

| Ataque | NC | | | Logos Recuperados | | |
|---|---|---|---|---|---|---|
| | Banda r | Banda g | Banda b | Banda r | Banda g | Banda b |
| Sin ataque | 1.0 | 1.0 | 1.0 |  |  |  |
| Corte 25 % horizontal | 1.0 | 1.0 | 1.0 |  |  |  |

| Ataque (cont.) | NC (cont.) | | | Logos Recuperados (cont.) | | |
|---|---|---|---|---|---|---|
| | **Banda r** | **Banda g** | **Banda b** | **Banda r** | **Banda g** | **Banda b** |
| Corte 25 % vertical | 1.0 | 1.0 | 1.0 |  |  |  |
| Corte 50 % horizontal | 1.0 | 1.0 | 1.0 |  |  |  |
| Corte 50 % vertical | 1.0 | 1.0 | 1.0 |  |  |  |
| Corte contorno | 1.0 | 1.0 | 1.0 |  |  |  |
| Llaves incorrecta | 0.1795 | 0.1918 | 0.1959 |  |  |  |
| Giro 90°derecha | 0.4653 | 0.5183 | 0.5142 |  |  |  |
| Giro 90 °izquierda | 0.4816 | 0.4489 | 0.4367 |  |  |  |
| Giro 180° | 0.5265 | 0.5387 | 0.5510 |  |  |  |

Cuadro 4.15: Baboon: Ataques de corte y rotación.

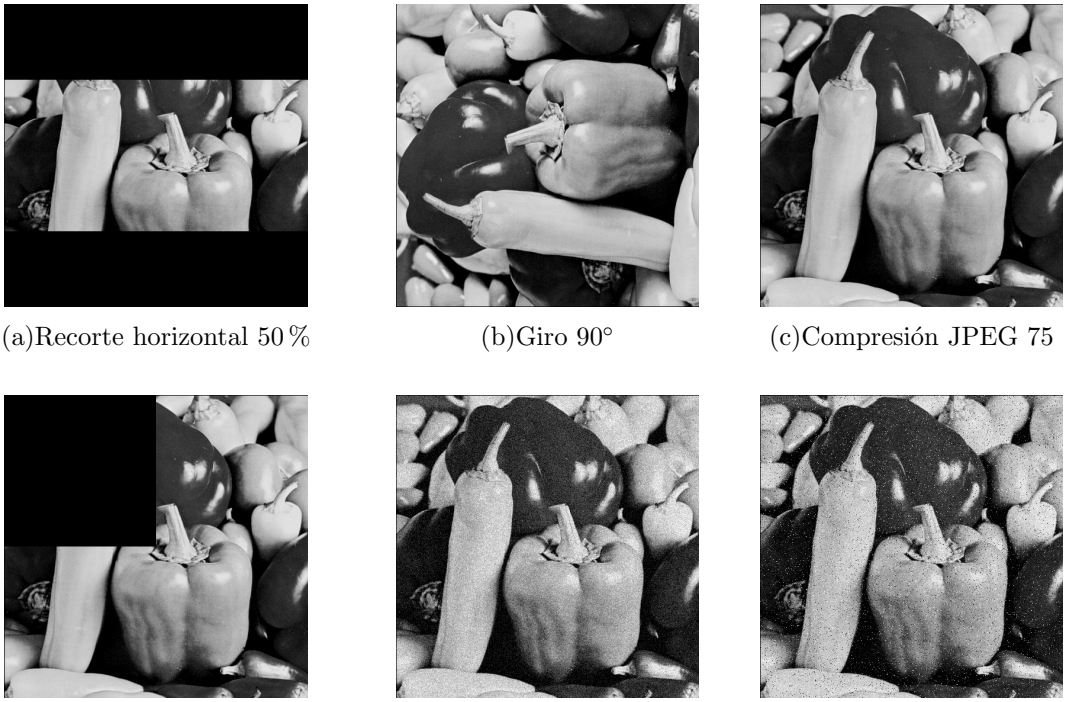
| Ataque | NC | | | Logos Recuperados | | |
|---|---|---|---|---|---|---|
| | **Banda r** | **Banda g** | **Banda b** | **Banda r** | **Banda g** | **Banda b** |
| Compresion JPEG (80) | 0.9632 | 0.9795 | 0.9387 |  |  |  |

| Ataque (cont.) | NC (cont.) | | | Logos Recuperados (cont.) | | |
|---|---|---|---|---|---|---|
| | Banda r | Banda g | Banda b | Banda r | Banda g | Banda b |
| Compresion JPEG (85) | 0.9877 | 0.9959 | 0.9673 |  |  |  |
| Compresion JPEG (90) | 1.0 | 1.0 | 0.9918 |  |  |  |
| Ruido Gaussiano ($\sigma = 5$) | 1.0 | 1.0 | 1.0 |  |  |  |
| Ruido Gaussiano ($\sigma = 10$) | 0.9918 | 0.9918 | 0.9918 |  |  |  |
| Ruido Gaussiano ($\sigma = 15$) | 0.9061 | 0.9142 | 0.9224 |  |  |  |
| Ruido S&P (Gamma = 0.5) | 1.0 | 0.9959 | 0.9918 |  |  |  |
| Ruido S&P (Gamma = 2.5) | 0.9306 | 0.9142 | 0.9224 |  |  |  |
| Ruido S&P (Gamma = 5) | 0.7673 | 0.7755 | 0.8122 |  |  |  |

Cuadro 4.16: Baboon: Ataques de compresión y adición de ruido.

**Peppers: 512x512 píxeles, escala de grises. Marca de agua: Logo 2**

Éstos son algunos ejemplos de los ataques que se hicieron a la imagen marcada con el fin de probar la robustez del esquema de marcas de agua.



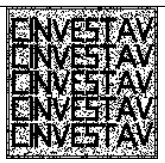
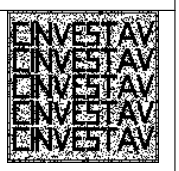
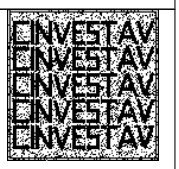
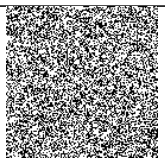
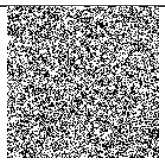
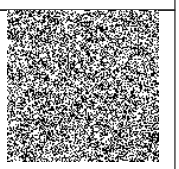
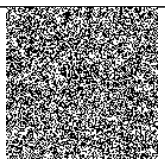
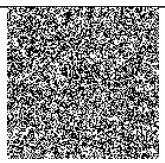
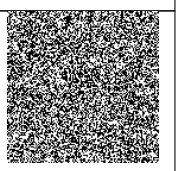
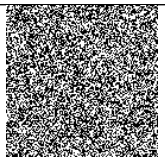
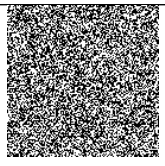
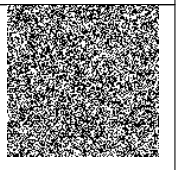
(a)Recorte horizontal 50 % (b)Giro 90° (c)Compresión JPEG 75

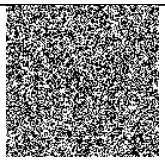
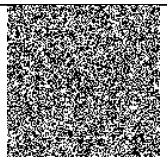
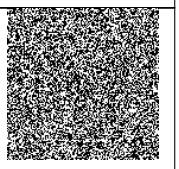
(d)Corte Horizontal-Vertical 25 % (e)Ruido Gaussiano($\sigma = 20$) (f)Ruido S&P (Gamma = 5)

Figura 4.21: Diferentes ataques a la imagen marcada

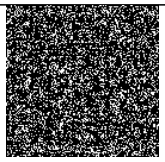
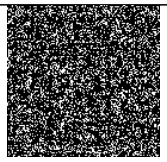
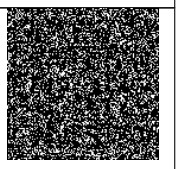
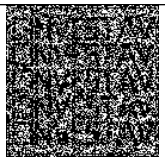
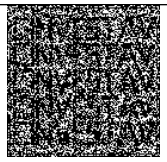
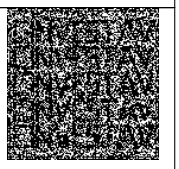
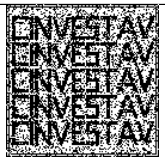
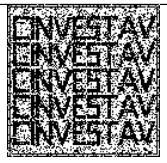
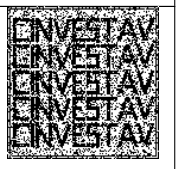
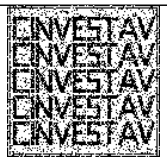
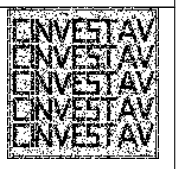
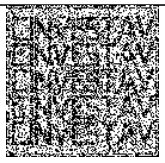
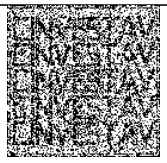
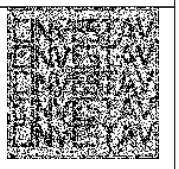
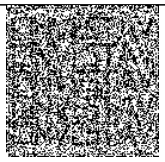
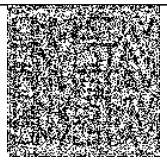
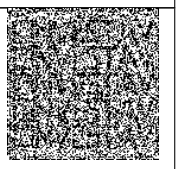
En seguida se muestran los resultados obtenidos en cada uno de los ataques realizados. La primera tabla abarca los ataques de corte y rotación (Tabla 4.18), mientras que la segunda contiene los ataques de compresión y adición de ruido (Tabla 4.19).

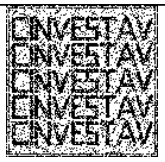
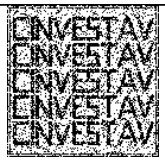
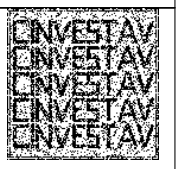
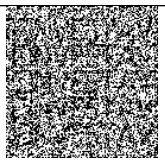
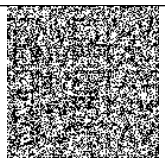
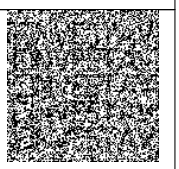
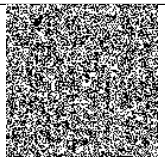
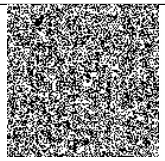
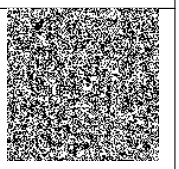
| **Ataque** | **NC** | | | **Logos Recuperados** | | |
|---|---|---|---|---|---|---|
| | **Banda r** | **Banda g** | **Banda b** | **Banda r** | **Banda g** | **Banda b** |
| Sin ataque | 0.9971 | 0.9971 | 0.9971 |  |  |  |

| Ataque (cont.) | NC (cont.) | | | Logos Recuperados (cont.) | | |
|---|---|---|---|---|---|---|
| | Banda r | Banda g | Banda b | Banda r | Banda g | Banda b |
| Corte 25 % horizontal | 0.9312 | 0.9312 | 0.9312 |  |  |  |
| Corte 25 % vertical | 0.9361 | 0.9361 | 0.9361 |  |  |  |
| Corte 50 % horizontal | 0.7387 | 0.7387 | 0.7387 |  |  |  |
| Corte 50 % vertical | 0.7530 | 0.7530 | 0.7530 |  |  |  |
| Corte horizontal-vertical 25 % | 0.9344 | 0.9344 | 0.9344 |  |  |  |
| Llaves incorrecta | 0.5680 | 0.5680 | 0.5680 |  |  |  |
| Giro 90°derecha | 0.4888 | 0.4888 | 0.4888 |  |  |  |
| Giro 90 °izquierda | 0.4950 | 0.4950 | 0.4950 |  |  |  |

| **Ataque (cont.)** | **NC (cont.)** | | | **Logos Recuperados (cont.)** | | |
|---|---|---|---|---|---|---|
| | **Banda r** | **Banda g** | **Banda b** | **Banda r** | **Banda g** | **Banda b** |
| Giro 180° | 0.4763 | 0.4763 | 0.4763 |  |  |  |

Cuadro 4.18: Peppers: Ataques de corte y rotación.

| **Ataque** | **NC** | | | **Logos Recuperados** | | |
|---|---|---|---|---|---|---|
| | **Banda r** | **Banda g** | **Banda b** | **Banda r** | **Banda g** | **Banda b** |
| Compresion JPEG (80) | 0.3268 | 0.3268 | 0.3268 |  |  |  |
| Compresion JPEG (85) | 0.4558 | 0.4558 | 0.4558 |  |  |  |
| Compresion JPEG (90) | 0.7316 | 0.7316 | 0.7316 |  |  |  |
| Ruido Gaussiano ($\sigma = 5$) | 0.8598 | 0.8598 | 0.8598 |  |  |  |
| Ruido Gaussiano ($\sigma = 10$) | 0.7003 | 0.7003 | 0.7003 |  |  |  |
| Ruido Gaussiano ($\sigma = 15$) | 0.6404 | 0.6404 | 0.6404 |  |  |  |

| Ataque (cont.) | NC (cont.) | | | Logos Recuperados (cont.) | | |
|---|---|---|---|---|---|---|
| | Banda r | Banda g | Banda b | Banda r | Banda g | Banda b |
| Ruido S&P (Gamma = 0.5) | 0.8303 | 0.8303 | 0.8303 |  |  |  |
| Ruido S&P (Gamma = 2.5) | 0.6279 | 0.6279 | 0.6279 |  |  |  |
| Ruido S&P (Gamma = 5) | 0.5773 | 0.5773 | 0.5773 |  |  |  |

Cuadro 4.19: Peppers: Ataques de compresión y adición de ruido.

### 4.3.4. Análisis de resultados del esquema robusto

A continuación se dará un análisis más detallado de los resultados obtenidos del esquema robusto de marcado de agua, la principal característica de este tipo de esquemas es que la marca de agua prevalece a pesar de que la imagen marcada ha sido alterada, intencionalmente o no. Los esquemas robustos de marcas de agua son generalmente utilizados para comprobación de derechos de autor y entre los ataques típicos encontramos los geométricos (cortes, rotaciones, etc.), de compresión con pérdida y de adición de ruido.

Debido a su naturaleza, los esquemas robustos necesitan que la marca de agua sea insertada con mas fuerza, es por ello que se ocupa metodos con transformadas. Para este caso de estudio, el esquema robusto que se describió anteriormente utiliza la DCT y el factor que mide la fuerza de inserción de la marca de agua es *delta*, en esta implementación se está utilizando $delta = 3$.

Otro aspecto que debemos tomar en cuenta es que se esta utilizando como marca de agua un logo en blanco y negro. Debido a las propiedades del esquema es posible insertar más de una vez dicha marca, todo depende del tamaño de la marca y de la imagen original. El número de veces que se puede insertar la marca se calcula mediante la formula 3.10, existe una relación directa entre el valor de este parámetro y la eficacia del esquema: entre mayor sea el número de copias de la marca de agua dentro de la imagen, se obtendrán mejores resultados.

Para medir el grado de robustez del esquema contra los diferentes ataques se optó por utilizar dos métricas: el reconocimiento visual y la correlación normalizada (NC). Se escogieron dos diferentes métricas porque el problema de decidir si una marca recuperada es válida o no, se vuelve difícil tomando en cuenta solo un criterio de decisión. Por el contrario, teniendo ambas métricas aseguramos una mejor decisión, ya que habrá veces que visualmente la marca no se pueda reconocer pero es entonces cuando se recurre a la NC.

Hablando mas específicamente del desempeño del algoritmo contra ataques geométricos podemos decir que para los ataques de corte se obtuvieron buenos resultados ya que en todos los casos se pudo reconocer la marca utilizando cualquiera de las métricas, desafortunadamente para los ataques de rotación no se tuvo el mismo éxito.

En cuanto a los ataques de adición de ruido, se probaron dos diferentes: ruido gaussiano y ruido sal y pimienta (S&P). El esquema mostró mejores resultados contra ruido gaussiano ya que es menos destructivo, su desempeño disminuyó ante el ruido S&P. En cuanto a los ataques de compresión con pérdida, se probaron niveles de compresión de 80 en adelante obteniendo buenos resultados.

Lo que no debemos olvidar es que si queremos obtener un buen desempeño del algoritmo contra la mayoria de los ataques, lo ideal es que se puedan insertar 16 o mas copias de la marca de agua dentro de la imagen. Un claro ejemplo de que los resultados mejoran bajo estas circunstancias es el último conjunto de pruebas de la sección anterior donde la imagen mide 512x512 píxeles y el logo 32x32 píxeles, lo cual nos da un total de 32 copias de la marca dentro de la imagen, fué en este grupo de pruebas donde se obtuvieron los mejores resultados.

# Capítulo 5

# Una aplicación de prueba para Android

En los capítulos anteriores se plantearon las bases para la implementación de dos esquemas de marca de agua en un dispositivo móvil: un esquema de marcado robusto y un esquema de marcado frágil. En este capítulo se presenta un aplicación de prueba desarrollada para Android, dicha aplicación se construyó utilizando las definiciones y estrategias descritas anteriormente.

Para comenzar se da una breve explicación del concepto y componentes principales de Android, posteriormente se presenta el dispositivo móvil que se utilizó para ejecutar la aplicación y finalmente la descripción de la aplicación de prueba.

## 5.1. Android para dispositivos móviles

Android constituye una pila de software pensada especialmente para dispositivos móviles y que incluye tanto un sistema operativo, como *middleware* y diversas aplicaciones de usuario. Representa la primera incursión seria de Google en el mercado móvil.

En general, una pila o plataforma de software es un elemento crucial en el desarrollo del mismo, ya que nos proporciona un marco de trabajo que permite crear nuevo software y que éste se pueda ejecutar sobre ella posteriormente. Lo anterior puede ser visto como un modelo de capas en dónde las plataforma de desarrollo funge como intermediario entre el hardware y las aplicaciones que se han desarrollado (ver Fig. 5.1). Las plataformas de desarrollo típicas incluyen un sistema operativo (S.O.), lenguajes de programación, sus correspondientes bibliotecas de funciones e interfaces gráficas (*User Interface* o UI).


Figura 5.1: Modelo de capas para desarrollo de software

Todas las aplicaciones para Android se programan en lenguaje Java y son ejecutadas en una máquina virtual especialmente diseñada para esta plataforma, que ha sido bautizada con el nombre de *Dalvik*. El núcleo de Android está basado en Linux 2.6 y es distribución libre, a los desarrolladores se les proporciona de forma gratuita un SDK y la opción de un plug-in para el entorno de desarrollo Eclipse que incluye todas las APIs necesarias para la creación de aplicaciones, así como un emulador integrado para su ejecución. Existe además disponible una amplia documentación de respaldo para este SDK.

El proyecto Android está capitaneado por Google y un conjunto de empresas tecnológicas agrupadas bajo el nombre de *Open Handset Alliance* (OHA). El objetivo principal de esta alianza empresarial (que incluye a fabricantes de dispositivos y operadores, con firmas tan relevantes como Samsung, LG, Telefónica, Intel o Texas Instruments, entre muchas otras) es el desarrollo de estándares abiertos para la telefonía móvil como medida para incentivar su desarrollo y para mejorar la experiencia del usuario.

Con Android se busca reunir en una misma plataforma todos los elementos necesarios que permitan al desarrollador controlar y aprovechar al máximo cualquier funcionalidad ofrecida por un dispositivo móvil (llamadas, mensajes de texto, cámara, agenda de contactos, conexión Wi-Fi, Bluetooth, aplicaciones ofimáticas, videojuegos, etc.), así como poder crear aplicaciones que sean verdaderamente portables, reutilizables y de rápido desarrollo. En otras palabras, Android quiere mejorar y estandarizar el desarrollo de aplicaciones para cualquier dispositivo móvil.

Existen algunas diferencias que hacen de Android una opción muy interesante para los fabricantes, y cómo no, para los usuarios y desarrolladores. A diferencia de sus competidores, Android es software libre, lo que permite que los fabricantes puedan usarlo sin necesidad de pagar. Por otra parte, al tener como base Linux, es fácilmente portable y adaptable a casi cualquier hardware. Android no es la primera plataforma de desarrollo móvil basado en Linux y que es software libre, Nokia abandera el proyecto *Maemo*, e incluso Ubuntu desarrolla *Ubuntu Mobile*, pero no parecen alcanzar la masa crítica necesaria [24].

### 5.1.1. Características

El diseño de Android cuenta, entre otras, con las siguientes características:

- Los componentes básicos de las aplicaciones se pueden sustituir fácilmente por otros.
- Cuenta con su propia máquina virtual, *Dalvik*, que interpreta y ejecuta código escrito en Java.
- Permite la representación de gráficos 2D y 3D.
- Almacenamiento de datos en SQLite.
- Servicio de localización GSM.
- Soporte para diferentes formatos de contenido multimeda: MPEG-4, H.264, MP3, AAC, OGG, AMR, JPEG, PNG, GIF.
- Conectividad (GSM/EDGE, CDMA, EV-DO, UMTS, Bluetooth y Wi-Fi).
- Soporte para hardware adicional: cámaras de video, pantallas táctiles, GPS, acelerómetros, entre otros.
- Mensajería (SMS y MMS).
- Navegador Web.
- Entorno de desarrollo que incluye: un emulador, herramientas de depuración, perfiles de memoria y funcionamiento, plugin para Eclipse IDE.

### 5.1.2. Arquitectura

En esta sección se dará una visión global de las capas que integran la arquitectura de Android. Cada una de éstas capas utiliza servicios ofrecidos por las anteriores, y a su vez ofrece ciertos servicios a las capas superiores [24, 25] (ver Fig.5.2).

- Aplicaciones: Es la capa superior de la pila, éste nivel incluye tanto las aplicaciones incluidas por defecto de Android como aquellas que el usuario vaya aadiendo posteriormente, ya sean de terceras empresas o de su propio desarrollo. Todas estas aplicaciones utilizan los servicios, las API y bibliotecas de los niveles que se encuentran debajo.
- Framework de aplicaciones: Los desarrolladores tienen acceso completo a las APIs del framework usado por las aplicaciones base. La arquitectura está diseñada para simplificar el reuso de componentes; cualquier aplicación puede publicar sus capacidades y cualquier otra puede hacer uso de esas capacidades (sujeto a reglas de seguridad del framework). Éste mismo mecanismo permite que los componentes sean reemplazados por el usuario. Este framework está formado por varios componentes, entre los cuales podemos encontrar:

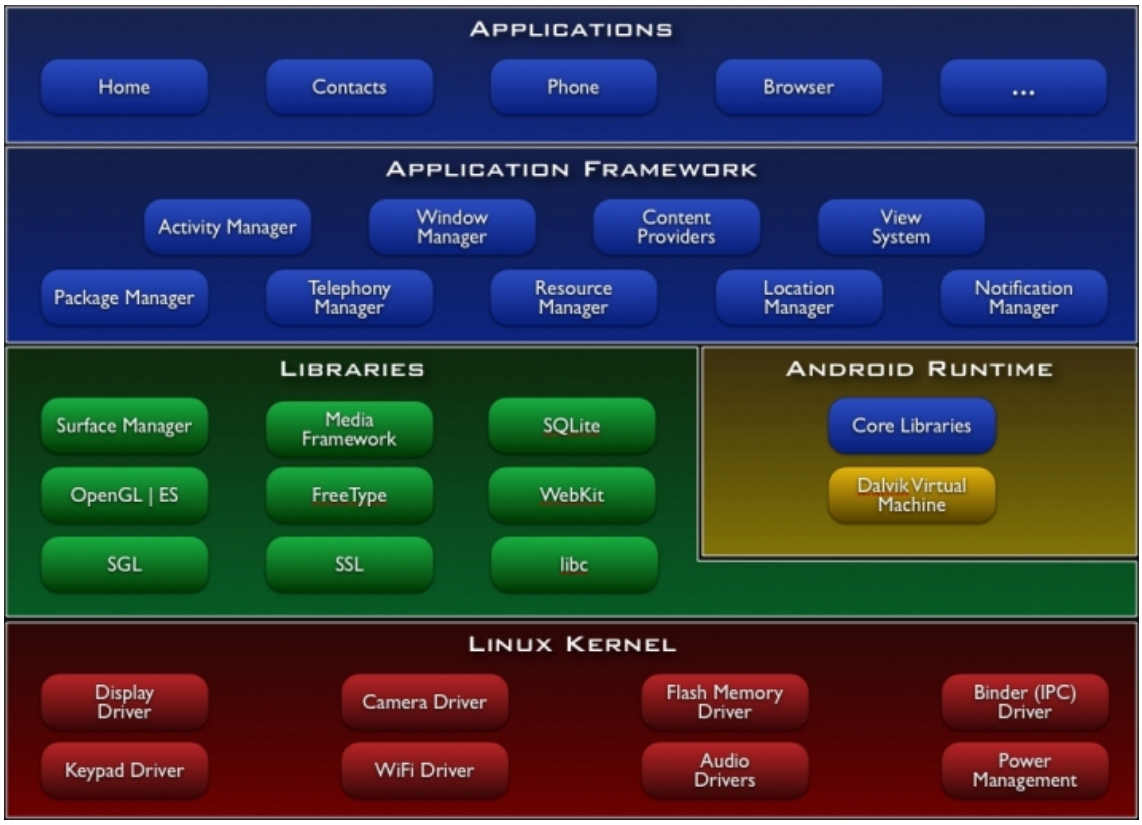



Figura 5.2: Arquitectura de Android

- Un extenso conjunto de Vistas tales como listas, cajas de texto, botones, entre otros.
- *Content Providers*, que permiten a las aplicaciones acceder a información de otras aplicaciones o compartir la propia.
- *Resource Manager*, que proporciona acceso a recursos que no son código como pueden ser gráficos, cadenas de texto, etc.
- *Notification Manager*, que permite a las aplicaciones mostrar alarmas personalizadas en la barra de estado.
- *Activity Manager*, que gestiona el ciclo de vida de las aplicaciones.

- Bibliotecas: Éstas han sido escritas utilizando C/C++ y proporcionan a Android la mayor parte de sus capacidades más caracterásticas. Junto con el núcleo basado en Linux, estas bibliotecas constituyen el corazón de Android. Algunas de ellas son:
  - La biblioteca *libc* incluye todas las cabeceras y funciones según el estándar del lenguaje C. Todas las demás bibliotecas se definen en este lenguaje.
  - La biblioteca *Surface Manager* es la encargada de componer los diferentes elementos de navegación de pantalla. Gestiona también las ventanas pertenecientes a las distintas aplicaciones.

- *OpenGL/SL y SGL* representan las bibliotecas gráficas y, por tanto, sustentan la capacidad gráfica de Android.
- La biblioteca *Media Libraries* proporciona todos los códecs necesarios para el contenido multimedia soportado en Android (vídeo, audio, imágenes estáticas y animadas, etc.)
- *FreeType*, permite trabajar de forma rápida y sencilla con distintos tipos de fuentes.
- La biblioteca *SSL* posibilita la utilización de dicho protocolo para establecer comunicaciones seguras.
- A través de la biblioteca *SQLite*, Android ofrece la creación y gestión de bases de datos relacionales, pudiendo transformar estructuras de datos en objetos fáciles de manejar por las aplicaciones.
- La biblioteca *WebKit* proporciona un motor para las aplicaciones de tipo navegador, y forma el núcleo del actual navegador incluido por defecto en la plataforma Android.

- Runtime de Android: Al mismo nivel que las bibliotecas de Android se sitúa el entorno de ejecución. Éste lo constituyen las *Core Libraries*, que son bibliotecas con multitud de clases de Java, y la máquina virtual *Dalvik*. Dentro de la máquina virtual se ejecutan archivos en el formato *Dalvik* Executable (.dex), el cual está optimizado para dispositivos con pocos recursos en cuanto a memoria se refiere.

- Núcleo de linux: Android utiliza el núcleo de Linux 2.6 como una capa de abstracción para el hardware disponible en los dispositivos móviles. Esta capa contiene los drivers necesarios para que cualquier componente de hardware pueda ser utilizado por las aplicaciones de la capa superior.

### 5.1.3. Componentes de una aplicación

Todas las aplicaciones en Android pueden descomponerse en cuatro tipos de bloques o componentes principales: *Activitity*, *Broadcast Intent Receiver*, *Service* y *Content Provider*. Cada aplicación de Android será una combinación de uno o más de estos componentes.

Los componentes de una aplicación, deberán ser declarados de forma explícita en un archivo con formato XML denominado AndroidManifest.xml [24, 25], junto a otros datos asociados como nombre de la aplicación, versión, valores globales, clases que implementa, datos que puede manejar, permisos, etc. Este archivo es básico en cualquier aplicación en Android y permite al sistema desplegar y ejecutar correctamente la aplicación.

A continuación se detallan los cuatro tipos de componentes en los que puede dividirse una aplicación para Android [25].

### Activity

Sin duda es el componente más habitual de las aplicaciones para Android. Un componente Activity refleja una determinada actividad llevada a cabo por una aplicación, y que lleva asociada típicamente una ventana o interfaz de usuario; es importante señalar que no contempla únicamente el aspecto gráfico, sino que éste forma parte del componente Activity a través de vistas representadas por clases como *View* y sus derivadas. Este componente se implementa mediante la clase de mismo nombre *Activity*.

Muy vinculado a este componente se encuentran los *Intents*, una interesante novedad introducida por Android. Un *Intent* consiste básicamente en la voluntad de realizar alguna acción, generalmente asociada a unos datos. Lanzando un *Intent*, una aplicación puede delegar el trabajo en otra, de forma que el sistema se encarga de buscar qué aplicación entre las instaladas es la que puede llevar a cabo la acción solicitada. Por ejemplo, abrir una URL en algún navegador web, o escribir un correo electrónico desde algún cliente de correo.

### Broadcast Intent Receiver

Un componente *Broadcast Intent Receiver* se utiliza para iniciar alguna otra acción dentro de la aplicación actual cuando un determinado evento se produzca (generalmente, abrir un componente *Activity*). Por ejemplo, una llamada entrante o un SMS recibido. No tiene interfaz de usuario asociada, pero puede utilizar el API *Notification Manager*, mencionada anteriormente, para avisar al usuario del evento producido a través de la barra de estado del dispositivo móvil. Este componente se implementa a través de una clase de nombre *BroadcastReceiver*.

Para que *Broadcast Intent Receiver* funcione, no es necesario que la aplicación en cuestión sea la aplicación activa en el momento de producirse el evento. El sistema lanzará la aplicación si es necesario cuando el evento monitorizado tenga lugar.

### Service

Un componente *Service* representa una aplicación ejecutada sin interfaz de usuario, y que generalmente tiene lugar en segundo plano mientras otras aplicaciones (con interfaz) son las que están activas en la pantalla del dispositivo. Un ejemplo típico de este componente es un reproductor de música. La interfaz del reproductor muestra al usuario las distintas canciones disponibles, así como los típicos botones de reproducción, pausa, volumen, etc. En el momento en el que el usuario reproduce una canción, ésta se escucha mientras se realiza alguna otra acción. Este elemento está implementado por la clase de mismo nombre *Service*.

**Content Provider**

Con el componente *Content Provider*, cualquier aplicación en Android puede almacenar datos en un fichero, en una base de datos SQLite o en cualquier otro elemento. Además, estos datos pueden ser compartidos entre distintas aplicaciones. Una clase que implemente el componente *Content Provider* contendrá una serie de métodos que permiten almacenar, recuperar, actualizar y compartir los datos de una aplicación. Existe una colección de clases para distintos tipos de gestión de datos en el paquete android.provider.

## 5.2. El dispositivo de prueba

Es importante mencionar que la implementación de los algoritmos se hizo tanto en un dispositivo móvil como en una computadora de escritorio esto con el fin de comprobar el correcto funcionamiento de los esquemas de marcas de agua previamente descritos.

Aunque se cuenta con ambas implementaciones en este capítulo solo se reportan los resultados obtenidos en el dispositivo móvil, ya que es la motivación central del presente trabajo.

Es común escuchar o leer que los dispositivos móviles, en comparación con las llamadas computadoras de escritorio, tienen restricciones en cuanto a capacidad de almacenamiento y procesamiento, debido a que dichos dispositivos vienen equipados con procesadores de baja frecuencia. Para ejecutar los esquemas de marcas de agua, que se han descrito anteriormente, se utilizó un dispositivo móvil con las siguientes características:

- Motorola
- Sistema Operativo Android 2.1
- Procesador a 600Mhz
- Pantalla Multi-Táctil TFT Capacitiva de 2.8" con resolución 320x240 píxeles
- Memoria interna de 150Mb
- 512Mb en RAM
- *MicroSD* de 2GB, ampliable hasta 32GB
- Cámara de 3.15 MPx
- Batería de 1130 mAh

## 5.3. Descripción de la aplicación

Para poder verificar el funcionamiento de los esquemas de marcado de agua previamente descritos, se desarrolló una aplicación para el dispositivo móvil que se mencionó anteriormente. A lo largo de esta sección se presentan y explican sus principales características y funcionalidades. Posteriormente se muestran los diagramas de paquetes y de clases que se utilizaron para el diseño de la misma, finalmente se presentan los tiempos de ejecución para cada uno de los esquemas de marca de agua.

### 5.3.1. Requerimientos

Debido a que el principal objetivo de la aplicación es probar el funcionamiento de los esquemas de marca de agua sobre imágenes digitales, la aplicación debe cumplir con los siguientes requerimientos:

- Permitir al usuario elegir qué esquema de marcado quiere utilizar: el esquema frágil o el esquema robusto.
- El usuario podrá elegir entre la operación de marcar/insertar o extraer/verificar la marca de agua para cada uno de los esquemas de marcado.
- Conceder al usuario la opción de cargar una foto del álbum o tomar una nueva utilizando la cámara del dispositivo.
- Cubrir los requisitos que cada uno de los esquemas de marcado necesita para su buen funcionamiento, en específico:
    - *Esquema frágil de marca de agua*
        - Tener un área para mostrar la imagen que se va a marcar, en caso de que se esté realizando la operación de marcar/insertar la marca de agua.
        - Tener un espacio para ingresar la llave del esquema tanto para el proceso de marcar/insertar como para el proceso de extraer/verificar la marca de agua.
        - Solicitar al usuario un nombre para la imagen marcada, en caso de que se esté realizando la operación de marcar/insertar la marca de agua.
        - Para la operación de extraer/verificar, tener un área para mostrar la imagen marcada.
        - Solicitar al usuario un nombre para la imagen recuperada, cuando se esté realizando la operación de extraer/verificar la marca de agua.
        - Presentar al usuario el resultado de la operación de extraer/verificar la marca de agua.
        - Indicar al usuario dónde se almacenáron las imágenes (marcada y recuperada).

- *Esquema robusto de marca de agua*
  - Tener un área para mostrar la imagen que se va a marcar y el logo que se va a usar como marca de agua, en caso de que se esté realizando la operación de marcar/insertar la marca de agua.
  - Tener un espacio para ingresar las llaves que necesita el esquema tanto para el proceso de marcar/insertar como para el proceso de extraer/verificar la marca de agua.
  - Permitir al usuario elegir el valor del parámetro *delta* en el proceso de marcar/insertar la marca de agua.
  - Solicitar al usuario un nombre para la imagen marcada, en caso de que se esté realizando la operación de marcar/insertar la marca de agua.
  - Indicar al usuario dónde se almacenó la imagen marcada.
  - Para la operación de extraer/verificar, tener un área para mostrar la imagen marcada y el logo con el que se va a hacer la comparación.
  - Presentar al usuario el logo recuperado y el valor de la métrica (NC) para cada una de las bandas de color (R,G,B).

### 5.3.2. Diseño

La etapa de diseño consiste en traducir los requerimientos del sistema a una representación de software. El diseño es el primer paso en la fase de desarrollo de cualquier producto o sistema de software. En esta sección se presentan el diagrama de paquetes utilizado para el desarrollo de la aplicación, posteriormente se presenta el diagrama de clases de ambos esquemas de marcado y finalmente se presentan las interfaces de usuario que conforman la aplicación.

#### Diagrama de paquetes

El diagrama de paquetes muestra los bloques generales en los que está dividida la aplicación, así como las dependencias y relaciones entre ellos. Como se puede apreciar en la Fig. 5.3, la aplicación está compuesta por tres paquetes:

- El paquete **cinvestav.watermarking.algoritmos**: contiene la implementación de los esquemas de marcado de agua. Esta compuesto por dos clases, la clase *Fragil* y la clase *Robusto*.
- El paquete **cinvestav.watermarking.utilerias**: contiene las clases que complementan la funcionalidad de los esquemas de marcado. Entre las clases que lo componen están: *DCT*, *Permutacion*, *ImagenMatriz*, *ImagenVector*, *Metricas*, entre otras.
- El paquete **cinvestav.watermarking.view**: esta formado por las clases que controlan la vista de la aplicación, algunas de ellas son: *MainActivity*, *MarkActivity*, *FragilResultActivity*, *RobustResultActivity*, entre otras.

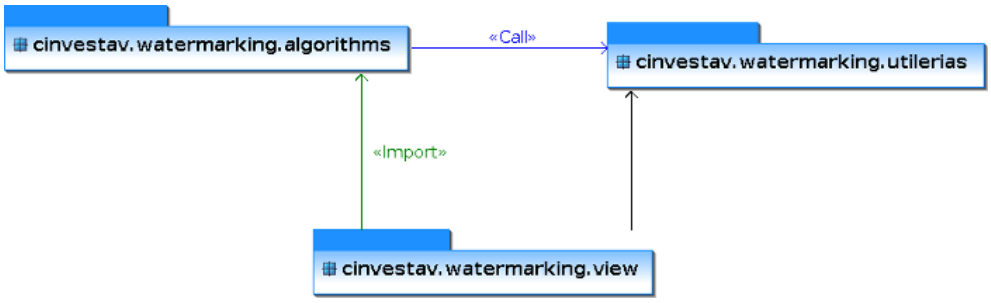



Figura 5.3: Diagrama de paquetes utilizado en la implementación

## Diagrama de clases

Los diagramas de clases presentan las clases que serán utilizadas por los esquemas de marcado dentro de la aplicación y las relaciones que existen entre ellas. A continuación se presenta el diagrama de clases que representa al esquema frágil de marca de agua (Fig. 5.4) y el diagrama de clases que representa al esquema robusto de marca de agua (Fig. 5.5).

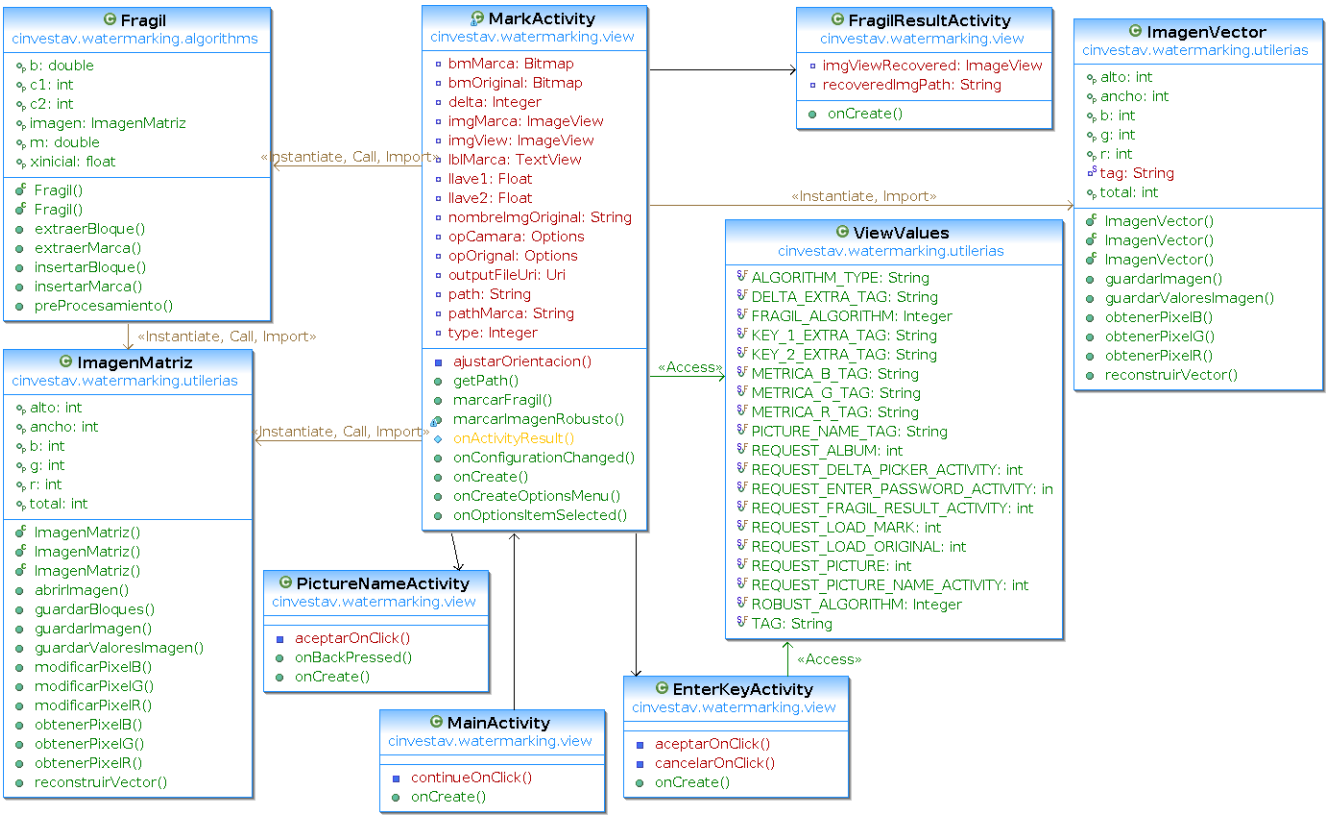



Figura 5.4: Diagrama de clases para el esquema de marcado frágil

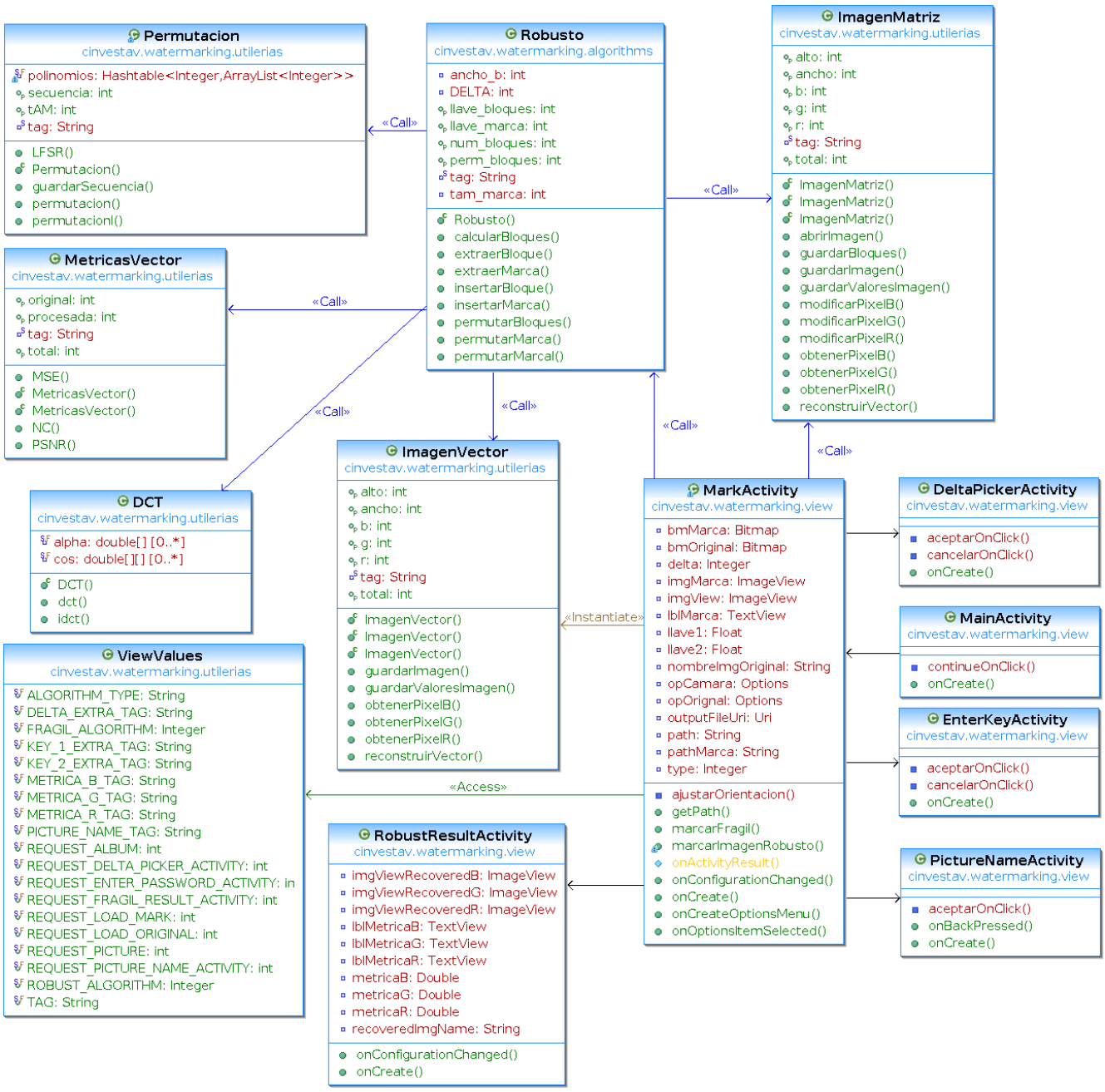



Figura 5.5: Diagrama de clases para el esquema de marcado robusto

### 5.3.3. Intefaces de usuario

Para cubrir cada uno de los requerimientos de la aplicación, se diseñaron las siguientes interfaces de usuario:

#### Interfaz principal

Esta formada por dos controles de selección (*Spinners*) (Fig. 5.6(a)), los cuales permiten al usuario seleccionar tanto el esquema de marcado (frágil o robusto, Fig. 5.6(b)) como la operación que quieren realizar (marcar/insertar o extraer/verificar la marca de agua, Fig. 5.6(c)).


(a)

(b) (c)

Figura 5.6: Interfaz principal de la aplicación

#### Interfaces del esquema frágil

Para que el usuario pueda aplicar las dos operaciones del esquema de marcado frágil a una imagen, se cuenta con una interfaz por cada operación. La primera que se describe es la del proceso de marcar/insertar, consta de un área para mostrar la imagen original (*ImageView*) y un menú que le permite al usuario elegir entre las siguientes opciones (Fig. 5.7(a)):

- *Capturar imagen*: inicia la cámara del dispositivo y permite al usuario tomar una fotografía.

- *Seleccionar imagen*: muestra el albúm de las imagenes que ya están almacenadas en el dispositivo y de las cuales el usuario puede seleccionar una.

- *Ingresar llave*: permite al usuario ingresar la llave que inicializa el proceso de marcado (Fig. 5.7(b)).

- *Marcar*: inicia el proceso de marcado de la imagen que se ha capturado o elegido previamente, también se le pide al usuario un nombre con el cual será almacenada la imagen marcada (Fig. 5.7(c)).

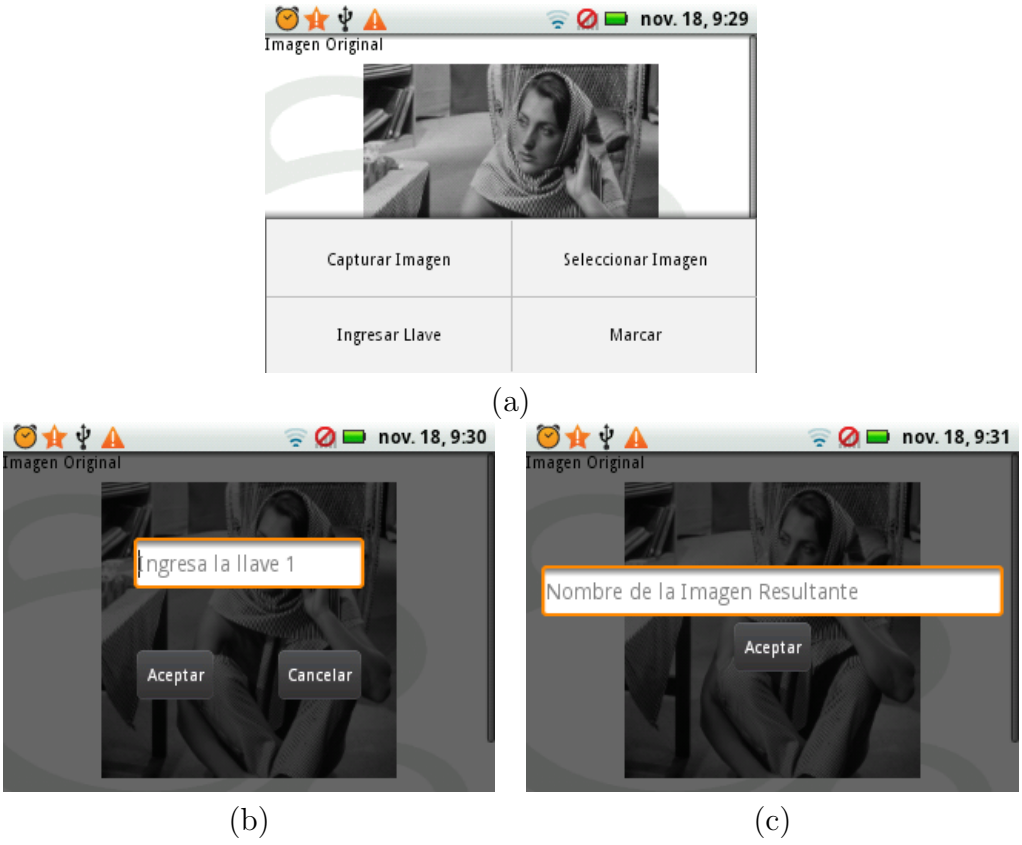



Figura 5.7: Interfaces del esquema frágil durante el proceso de marcar/insertar

Cuando el proceso de marcado ha terminado, le indica al usuario la ruta en la que se localiza la imagen marcada y lo dirige al menú principal (Fig. 5.8).

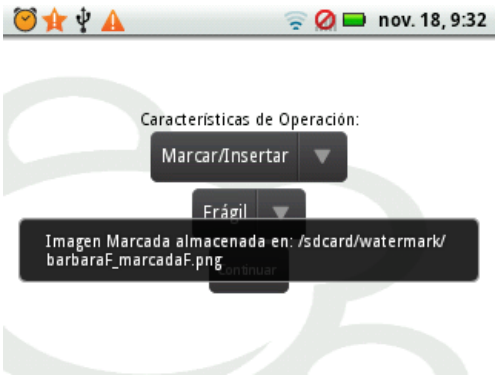



Figura 5.8: Termina el proceso de marcado e indica la ruta de la imagen marcada

A continuación se muestran las interfaces que involucran al proceso de extraer/verificar la marca de agua. La primera interfaz permite al usuario elegir la imagen a la que se le aplicará el proceso, esta acción se inicia cuando el usuario deja presionada la figura de la lupa (Fig. 5.9 (a)). Además de esto, contiene un menú con las siguientes opciones(Fig. 5.9(b)):

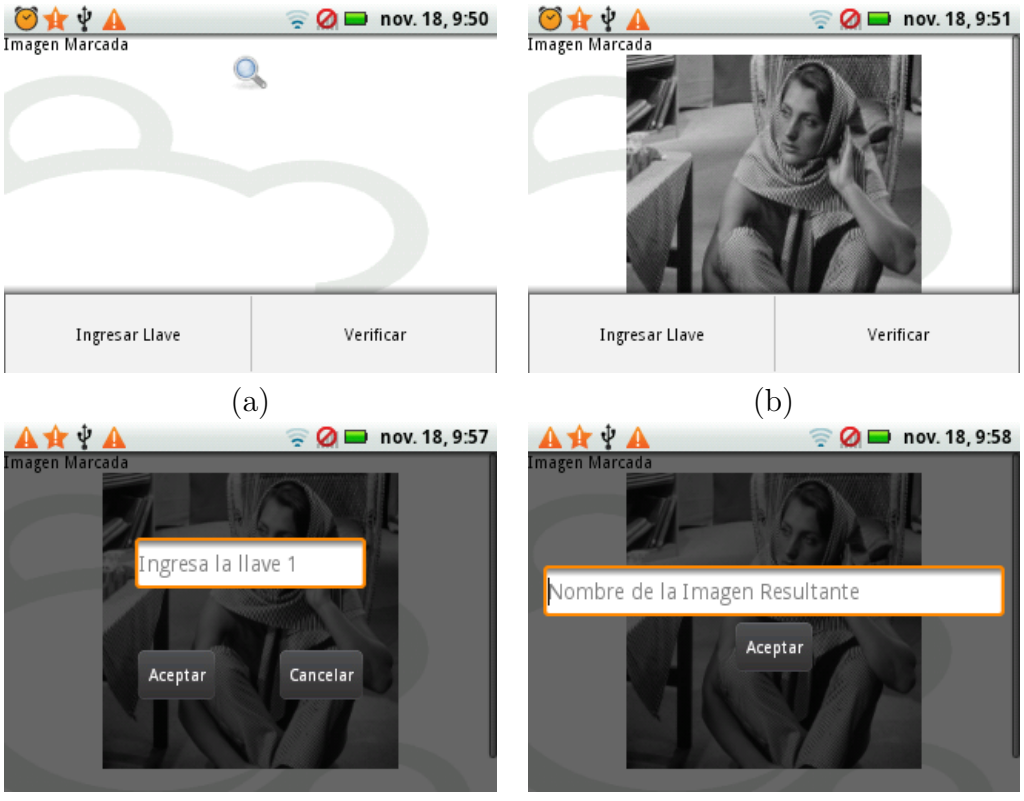



Figura 5.9: Interfaces del esquema frágil durante el proceso de marcar/insertar

- *Ingresar llave*: permite al usuario ingresar la llave que inicializa el proceso de extracción de la marca (Fig. 5.9(c)).
- *Verificar*: inicia el proceso de extracción de la marca de agua sobre la imagen que se ha elegido previamente, también se le pide al usuario un nombre con el cual será almacenada la imagen recuperada (Fig. 5.9(d)).

Al finalizar el proceso de extracción se le muestra al usuario la imagen recuperada y la ruta donde se encuentra almacenada (Fig. 5.10).


Figura 5.10: Termina el proceso de extracción y muestra la imagen recuperada

### Interfaces del esquema robusto

El esquema de marcado robusto tiene esta compuesto de dos procesos principales: el primero de ellos es marcar/insertar y el segundo extraer/verificar la marca de agua. Para que el usuario pueda aplicarlos a una imagen utilizando la aplicación se desarrollaron dos interfaces, una para cada proceso.


(a) (b)

Figura 5.11: Interfaces del esquema robusto durante el proceso de marcar/insertar

A continuación se describe la primera de ellas que corresponde al proceso de marcar/insertar, dicha interfaz permite al usuario seleccionar la marca de agua que se va utilizar y la muestra en la pantalla, para hacer la selección el usuario debe dejar presionado la figura de la lupa que aparece. Además de esto, la interfaz esta formada por un área donde se muestra la imagen original y un menú que contiene las siguientes opciones (Fig. 5.11):

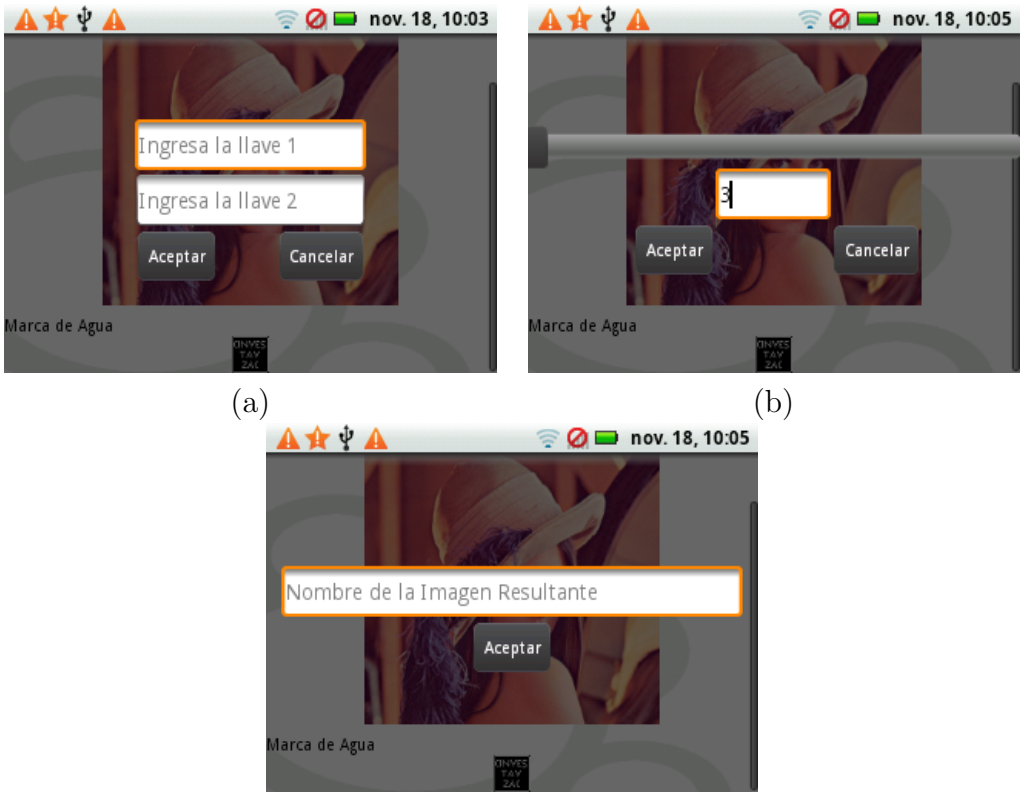



Figura 5.12: Interfaces del esquema robusto durante el proceso de marcar/insertar

- *Capturar imagen*: inicia la cámara del dispositivo y permite al usuario tomar una fotografía.

- *Seleccionar imagen*: muestra el albúm de las imagenes que ya están almacenadas en el dispositivo y de las cuales el usuario puede seleccionar una.

- *Ingresar llave*: permite al usuario ingresar las llaves que inicializan el proceso de marcado (Fig. 5.12(a)).

- *Seleccionar delta*: permite al usuario seleccionar el valor del parámetro *delta*, el cual indica la fuerza de inserción de la marca (Fig. 5.12(b)).

- *Marcar*: inicia el proceso de marcado de la imagen que se ha capturado o elegido previamente, también se le solicita al usuario un nombre con el cual será almacenada la imagen marcada (Fig. 5.12(c)).

Una vez que el proceso de marcado ha terminado, se le muestra al usuario la ruta donde se localiza la imagen marcada y lo dirige al menú principal (Fig. 5.13).

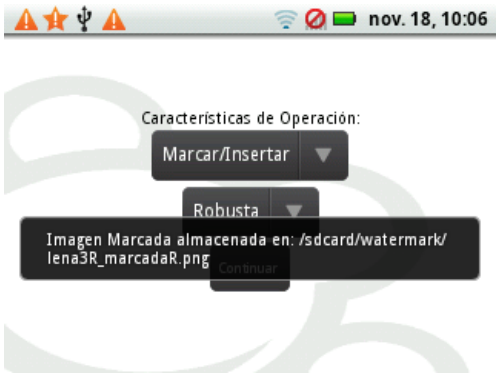



Figura 5.13: Termina el proceso de marcado e indica la ruta de la imagen marcada

En seguida se presenta la descripción de la interfaz vinculada al proceso de extraer/verificar la marca de agua. Ésta interfaz permite seleccionar la imagen marcada a la que se aplicará el proceso de extracción, se debe elegir también la marca de agua con la que se va a comparar las marcas recuperadas, ambas selecciones se hacen presionando la lupa que aparece en la interfaz (Fig. 5.14(a) y (b)).

El siguiente componente de la interfaz es un menú que permite al usuario seleccionar una opción de entre las que se mencionan a continuación:

- *Ingresar llave*: permite al usuario ingresar las llaves que inicializan el proceso de extracción de la marca (Fig. 5.14(c)).

- *Verificar*: inicia el proceso de extracción de la marca de agua en la imagen que se ha elegido previamente, también se le solicita al usuario un nombre con el cual serán almacenadas las marcas de agua recuperadas (Fig. 5.14(d)).

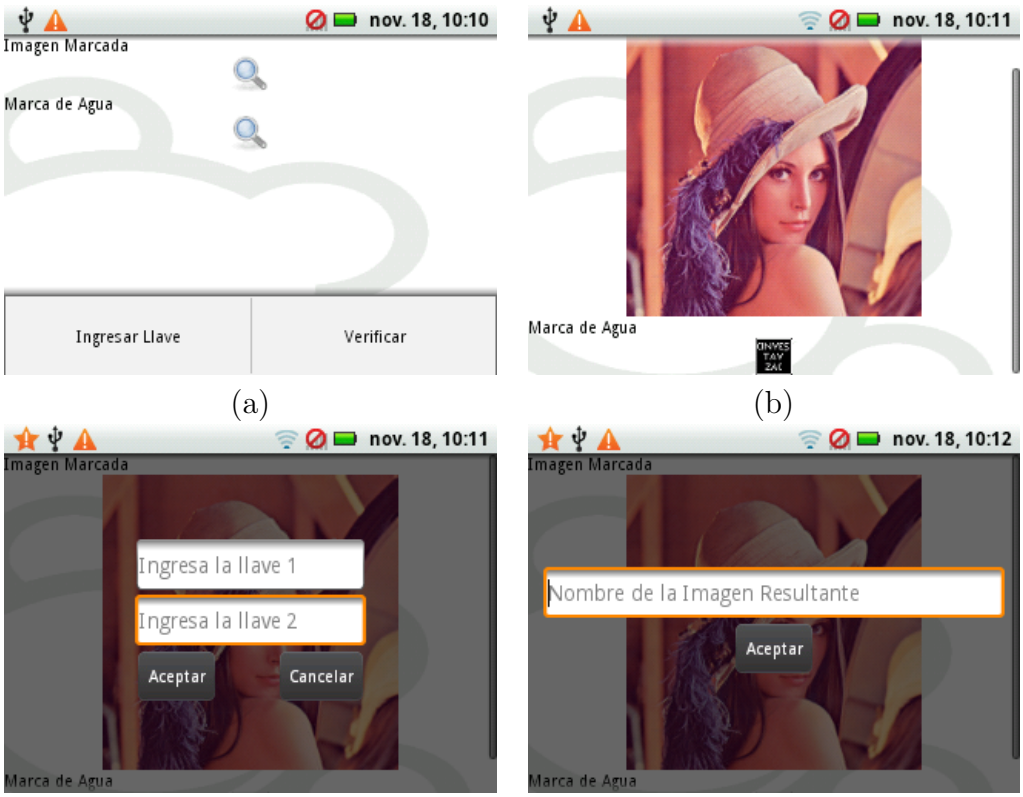



Figura 5.14: Interfaces del esquema robusto durante el proceso de extraer/verificar

Cuando finaliza el proceso de extracción se le muestra al usuario los logos recuperados en cada una de las bandas de color (R,G,B) y la métrica de comparación NC (Fig. 5.15).

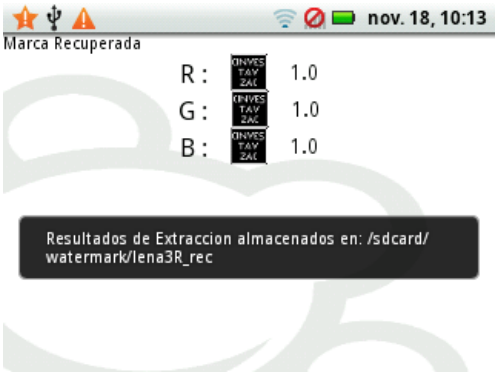


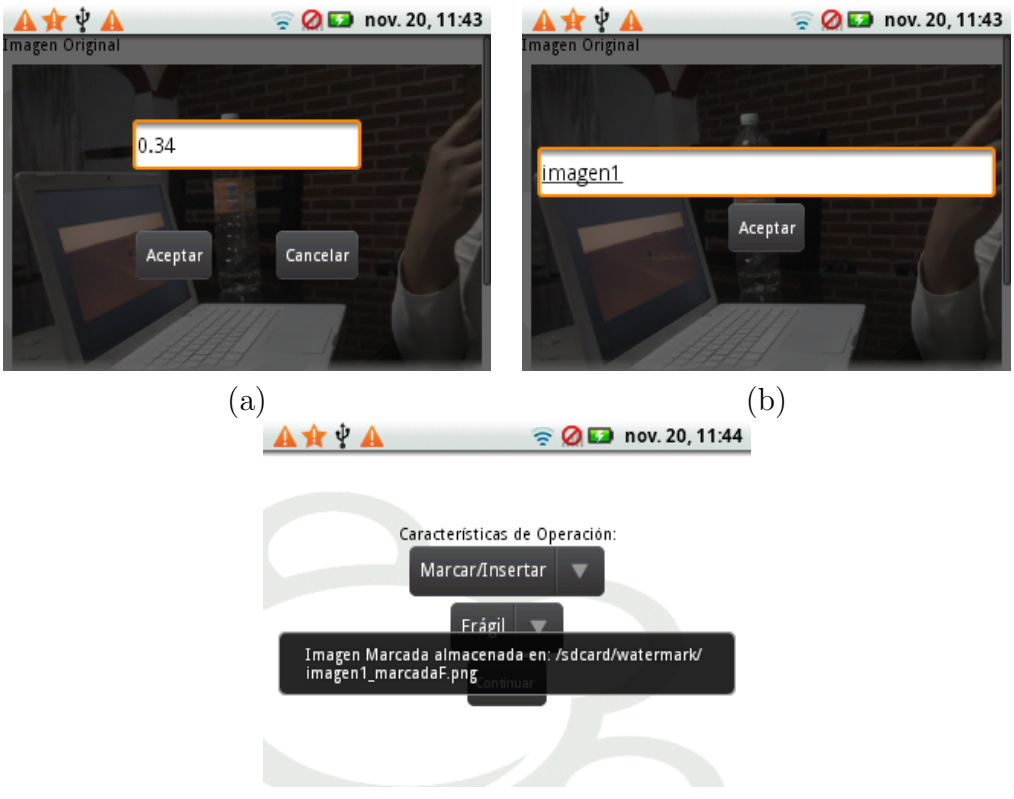

Figura 5.15: Termina el proceso de extracción: se muestran los logos recuperados y la métrica

### 5.3.4. Probando la aplicación

Para hacer una prueba más real de la aplicación, se seleccionó la opción de capturar una imagen con la cámara del dispositivo (Fig. 5.16). El esquema de marcado que se va a utilizar es el frágil .


(a) (b)

Figura 5.16: Seleccionar la opción de capturar una fotografía




(a) (b)

(c)

Figura 5.17: Proceso de marcado dentro del dispositivo móvil

Una vez que se ha capturado la imagen es necesario ingresar la llave del esquema de marcado (Fig. 5.17(a)) y asignar un nombre a la imagen marcada (Fig. 5.17(b)). Una vez que el proceso de marcado ha finalizado, aparce un mensaje indicandonos dónde quedó almacenada la imagen marcada (Fig. 5.17(c)).

Para comprobar el funcionamiento del esquema de marcado dentro del dispositivo móvil, es necesario ejecutar el proceso de extracción de la marca. Para ello es necesario seleccionar la imagen que ha sido marcada (Fig. 5.18(a)), ingresar la llave que se utilizó en el proceso de marcado (Fig. 5.18(b)) y dar un nombre para la imagen recuperada (Fig. 5.18(c)).

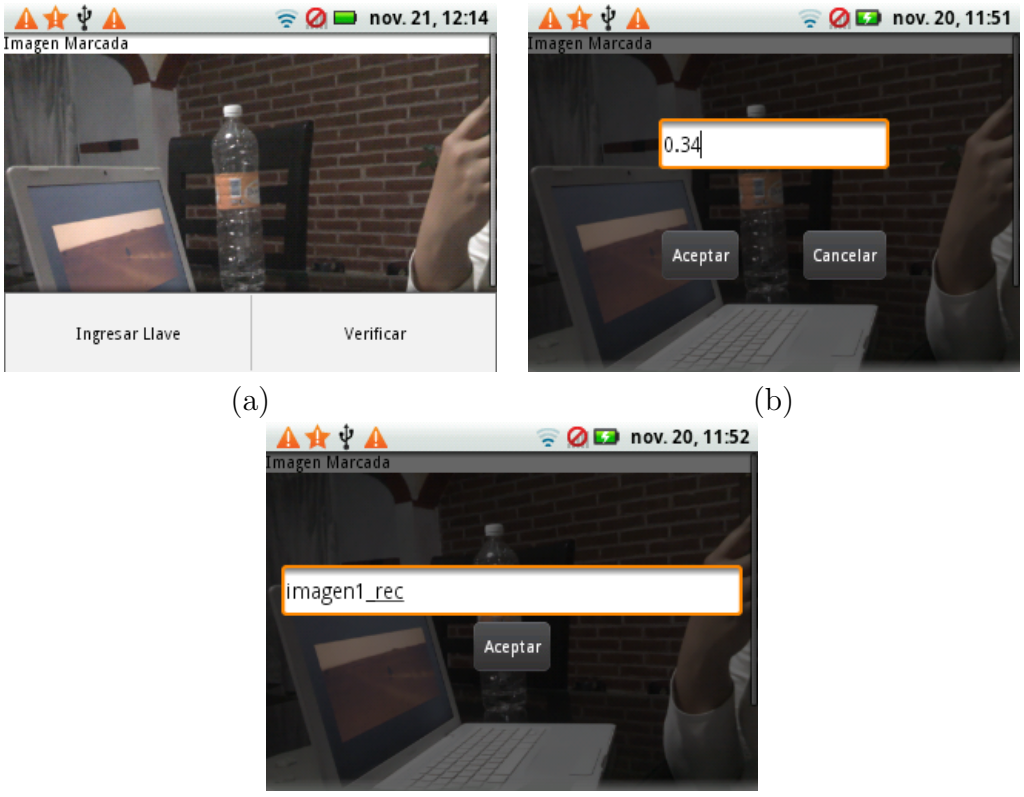



Figura 5.18: Proceso de extracción dentro del dispositivo móvil

Cuando finaliza el proceso de extracción nos muestra la imagen recuperada y su ruta de ubicación (Fig. 5.19). Como podemos observar la imagen recuperada es idéntica a la imagen original, esto significa que la imagen no ha sido alterada y por lo tanto se puede catalogar como auténtica.

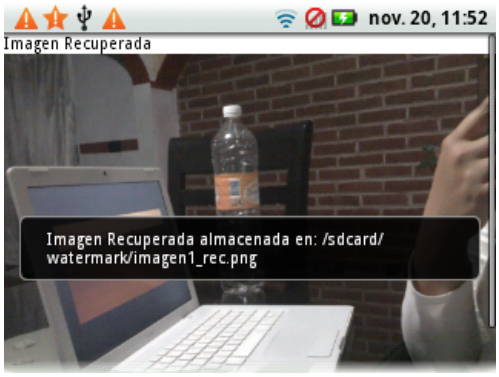



Figura 5.19: Resultado del proceso de extracción

En seguida se presentan los tiempos de ejecución para cada uno de los esquemas de marcado de agua usando el dispositivo móvil. La primera tabla muestra los resultados del esquema frágil (Tabla 5.1):

| Tamaño de imagen (px) | Proceso | Tiempo de ejecución (s) |
|---|---|---|
| 256x256 | Inserción | 0.777 |
| | Detección | 0.482 |
| 512x512 | Inserción | 2.939 |
| | Detección | 1.767 |
| 384x512 | Inserción | 2.057 |
| | Extracción | 2.383 |

Cuadro 5.1: Tiempos de ejecución del esquema de marcado frágil en el dispositivo móvil

Ahora se muestran los resultados para el esquema de marcado robusto (Tabla 5.2):

| Tamaño de imagen (px) | Proceso | Tiempo de ejecución (s) |
|---|---|---|
| 256x256 | Inserción | 3.716 |
| | Extracción | 1.853 |
| 512x512 | Inserción | 13.838 |
| | Extracción | 6.562 |
| 384x512 | Inserción | 10.857 |
| | Extracción | 4.814 |

Cuadro 5.2: Tiempos de ejecución del esquema de marcado robusto en el dispositivo móvil

Es importante destacar que los tiempos de ejecución obtenidos son considerablemente buenos si tomamos en cuenta las restricciones del dispositivo móvil y la cantidad de procesamiento que significan las operaciones realizadas dentro de cada esquema, éstos tiempos irán creciendo conforme aumente el tamaño de la imagen, aunque también es relevante mencionar que es posible mejorarlos.

# Capítulo 6

# Conclusiones y trabajo a futuro

A lo largo de este documento, se habló de dos diferentes esquemas de marcado de agua: un esquema frágil para la autenticación de imágenes y un esquema robusto para la protección de derechos de autor. Se proporcionaron las bases teóricas y la descripción detallada de cada uno de ellos, pero el objetivo principal de este trabajo consistió en implementar dichos esquemas en una plataforma móvil. Para realizar una implementación eficiente fue necesario hacer ciertas adaptaciones o mejoras a los esquemas.

Para probar que las adaptaciones realizadas no afectaran el funcionamiento de los esquemas de marcado se sometieron a una serie de pruebas que ayudarón a determinar su robustez. Finalmente, para aterrizar todas la ideas planteadas a lo largo de este trabajo se realizó una aplicación de prueba utilizando Android como plataforma de desarrollo, dicha aplicación tiene como base los esquemas de marcado.

En este capítulo se presentan las conclusiones resultantes del trabajo de investigación que se llevo a cabo y para finalizar se mencionan algunas propuestas que se pueden considerar como trabajo a futuro.

## 6.1. Conclusiones

El principal aporte de este trabajo consistió en verificar que a pesar de las restricciones de un dispositivo móvil, es posible implementar un esquema de marca de agua que se ejecuta en un lapso de tiempo aceptable. Para comprobar lo anterior se realizó la implementación de dos esquemas:

- **Esquema robusto de marca de agua**: se utiliza para la protección de derechos de autor, trabaja en el dominio de la frecuencia, es invisible y ciego.

- **Esquema frágil de marca de agua**: sirve para autenticar imagenes, es un esquema espacial, invisible y ciego.

El algoritmo robusto trabaja en el dominio de la frecuencia, es por esta razón que requiere de operaciones computacionales mas costosas, tal es el caso del cálculo de la DCT. Por otro lado, el esquema frágil que es espacial hace uso de operaciones mas sencillas, por lo que presenta un mejor desempeño en cuanto a tiempo de ejecución. Debido a las limitaciones de los dispositivos móviles en cuanto a capacidad de procesamiento y almacenamiento, es mejor optar por esquemas espaciales ya que representan un menor costo computacional.

No obstante, siempre es recomendable hacer un análisis más detallado de cada uno de los esquemas de marcado con el fin de encontrar bloques dentro del esquema que puedan ser optimizados. Para el esquema frágil se redujo la cantidad de procesamiento al dejar de lado el esquema piramidal que proponen los autores en [11], en su lugar se propuso utilizar una única capa de marcas de agua para toda la imagen original. La eficiencia del esquema robusto se mejoró al hacer una optimización en la DCT, la cual consistió en utilizar tablas de consulta para las *funciones bases del coseno* y la funcion $\alpha(u)$.

Otra de las características importantes en un esquema de marcado de agua es que sea ciego, es decir que para extraer o verificar la marca de agua no se necesite la imagen original. Esta cualidad ayuda a la condición de movilidad de las aplicaciones que se puedan generar utilizando esquemas de marca de agua.

Como se mencionó en el capítulo 3, el esquema frágil de marca de agua está basado en la teoría del caos. Esta característica lo hace susceptible al efecto mariposa y es necesario considerar los problemas que se pueden originar debido al manejo inadecuado de la precisión numérica (decimales). Durante la implementación del esquema de marcado se debe tener cuidado al manejar los decimales tanto en el proceso de inserción como en el proceso de extracción, ya que de lo contrario se pueden generar respuestas erroneas en el esquema. Para corregir este problema, basta con asegurarnos que se este utilizando el mismo número de decimales dentro de las operaciones en ambos procesos.

Los dos esquemas de marcado que se utilizaron en el desarrollo de este trabajo requieren dividir a la imagen original en bloques de $n \times n$ píxeles, ésto representa una gran desventaja porque limita el tamaño de imagenes que se pueden marcar, ya que las dimensiones de dichas imagenes tienen que ser múltiplo del tamaño de bloque. Específicamente, para el esquema robusto las dimensiones de las imagenes deben ser múltiplo de 8 por la DCT y para el esquema frágil deben ser múltiplo de 32 debido al tamaño de bloque seleccionado para la localización de regiones alteradas.

Finalmante, otra de las limitaciones presentes en el esquema robusto se origina debido al uso de los LFSR, ya que para poder contemplar todas las posibilidades en cuanto a tamaños de marca de agua y número de bloques dentro de una imagen, es necesario tener un banco de polinomios que contemple cada posibilidad, lo cual resulta bastante difícil.

## 6.2. Trabajo a futuro

Aunque los resultados obtenidos son bastante satisfactorios, siempre hay detalles que se pueden mejorar o aspectos que se pueden agregar. Algunos de ellos se listan a continuación:

- Para mejorar aún mas la eficiencia de los esquemas de marca de agua se puede utilizar el NDK, el cual permite hacer desarollo a mas bajo nivel, es decir, utilizando código nativo de android (C/C++) que se invoca desde la aplicación utilizando librerías JNI.

- Para el esquema robusto en lugar de utilizar un logo como marca de agua, considerar la opción de utilizar un código QR. Esto es posible ya que dicho código se puede descomponer como una secuencia binaria, lo cual se adapta perfectamente a los requerimientos del esquema.

- Hasta el momento la principal métrica de eficiencia de los esquemas de marcado dentro del dispositivo móvil, es el tiempo de ejecución de los mismos. Sin embargo, no estaría de más incluir el consumo de batería para medir el rendimiento de los esquemas.

- Cuando el esquema de marcado que se está utilizando necesita dividir a la imagen original en bloques cuadrados de $n \times n$ píxeles, se restringe el tamaño de dicha imagen ya que idealmente sus medidas deben ser múltiplos de $n$, esto con el fin de lograr que la división sea exacta. Para erradicar esta restricción es posible utilizar cierto "relleno", en los bloques que queden incompletos. Sin embargo en necesario estudiar que tipo de "relleno" conviene utilizar y si no afecta el funcionamiento de los esquemas de marcado.

- La implementación de los esquemas de marcado de agua, solo representa un bloque en el desarrollo de aplicaciones comerciales más elaboradas y que brinden más servicios de seguridad. Es necesario contar con la infraestructura adecuada para el buen manejo de las llaves de los esquemas de marcado y para la correcta administración de la información digital que se ha protegido. Algunas de las áreas de aplicación potenciales para este tipo de implementaciones son el *e-commerce* (comercio electrónico) y el *e-governance* (gobierno eleéctronico).

- Otro enfoque que se puede explorar dentro del uso de marcas de agua como medio de protección, son las marcas de agua en imágenes impresas. Es decir, el proceso de inserción se realizó sobre la imagen digital y posteriormente dicha imagen protegida se imprimió en algún lugar. El algoritmo de extracción debe ser capaz de detectar la marca si nosotros tomamos una fotografía a la imagen impresa. Este enfoque puede ser utilizado para proteger documentos oficiales, carteles, pósters, láminas, fotografías, en general cualquier información que se encuentre impresa.

# Apéndice A

# Teoría del Caos

A continuación se mencionan algunos de los aspectos mas relevantes en el desarrollo de la Teoría del Caos, la cual se fortalece día a día y ha encontrado cabida en muchos ambientes de investigación científica: economía, biología, matemáticas, computación, etc. Su carácter multidisciplinario ha contribuido en gran medida a su desarrollo [26].

El caos y los sistemas caóticos no implican necesariamente desorden o falta de estructura en el sentido literal y popular de la palabra. Éste concepto va mucho más allá, la teoría del caos es un campo de estudio relativamente nuevo que puede definirse como: el estudio cualitativo de la conducta no periódica e inestable en sistemas dinámicos deterministas y no-lineales [27].

Los sistemas no lineales son sistemas irregulares, altamente impredecibles, que se manifiestan en muchos ámbitos de la vida y la naturaleza, pero que no se puede decir que tengan comportamientos sin ley, dado que existen reglas que determinan su comportamiento, aunque éstas sean difíciles de conocer en muchas ocasiones.

## A.1. Efecto mariposa: sensibilidad a las condiciones iniciales

El primer experimentador del caos fue un meteorólogo llamado Edward Lorentz. En 1960, Lorenz usaba computadoras para ayudarse en la solución de ecuaciones matemáticas que modelaban la atmósfera de la Tierra. Al hacer un pronóstico meteorológico introdujo datos para varias variables y acabó con una predicción del futuro estado del tiempo.

Más tarde, queriendo aclarar algunos detalles, regresó a su predicción y reintrodujo los datos sobre las variables del sistema. La primera vez, introdujo los números hasta el sexto decimal. Para la segunda prueba, redondeó a solo tres decimales. Cuando regresó para checar los resultados de la segunda prueba encontró una predicción completamente distinta [27].

Lo que Lorentz descubrió es una de las características más relevantes de la teoría del caos: los sistemas dinámicos no lineales muestran una dependencia sensible a las condiciones iniciales. Este concepto se ilustra en la célebre noción del efecto de mariposa"[26, 27] (ver Fig. A.1),cuya metáfora que no se debe tomar tan a la ligera establece que: *una mariposa que bate sus alas en algún lugar del amazonas puede provocar, a través de efectos encadenados y multiplicados, un huracán en el norte de Europa a miles de kilómetros de distancia.*

Lo que se quiere enfatizar, con una imagen chocante, es la dependencia extrema a las condiciones iniciales en un sistema caótico.

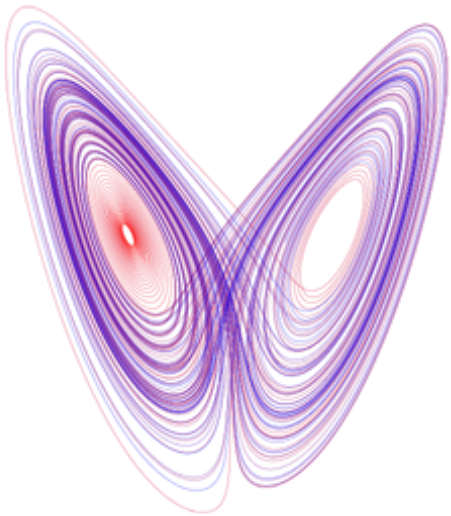


Figura A.1: Lorentz, 1960. Efecto mariposa

El esquema frágil de marca de agua propuesto en [11] y descrito en el Capítulo 3, basa su funcionamiento en la teoría del caos y utiliza una función no lineal llamada función caótica de Chebyshev, definida como:

$$f_{CHEB}(X_{i-1}) = tanh(C_1 \cdot X_{i-1}) - b \cdot tanh(C_2 \cdot X_{i-1}) \tag{A.1}$$

Se pueden encontrar mas detalles de esta función en el trabajo de Lai and Zhou [28].

# Apéndice B

# Registro de desplazamiento con retroalimentación lineal (LFSR)

Un registro de desplazamiento con retroalimentación lineal LFSR, puede ser utilizado para generar secuencias de números pseudoaleatorios utilizando bits que recorren un arreglo de celdas y que relacionan la salida con la entrada utilizando la operacin XOR; de ahí el nombre de realimentación lineal. Este tipo de generadores son máquinas de estados finitos, cuya salida depende del estado y la entrada presentes [22].

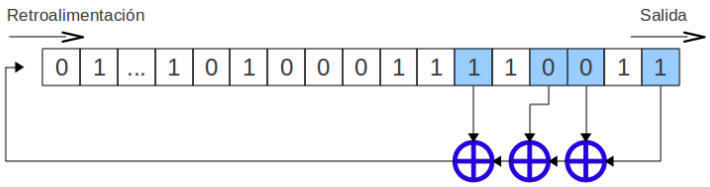



Figura B.1: Registro de desplazamiento con retroalimentación lineal (LFSR)

Un LFSR puede modelarse con base en un polinomio de la forma $p(x) = a_n x^n + a_{n-1}x^{n-1} + \ldots + a_2 x^2 + a_1 x + a_0$, el cual es conocido como polinomio característico [22, 23], donde $a \in [0, 1]$. Los coeficientes no nulos del polinomio característico representan las celdas del registro que actuarán en la retroalimentación.

Las principales características de un LFSR se listan a continuación[22, 23]:

1. El grado $n$ del polinomio característico indicará el número de celdas y se producirá una secuencia de periodo $2^n - 1$.

2. El polinomio debe ser primitivo; es decir, el polinomio no puede reducirse o factorizarse en el campo primitivo de característica dos.

3. Debe evitarse la secuencia nula

# Apéndice C

# La transformada discreta del coseno (DCT)

También denominada transformada del coseno, es la más ampliamente utilizada en aplicaciones de compresión de imágenes y vídeo. Las cualidades de la DCT permiten obtener altos índices de compresión a muy bajo coste, ya que esta transformada cuenta con una buena propiedad de compactación de energía.

Otra característica valiosa de la DCT es la decorrelación de coeficientes, que es un aspecto muy importante para compresión, ya que, el posterior tratamiento de cada uno de los coeficientes se puede realizar de forma independiente, sin pérdida de eficiencia en la compresión. La DCT está bastante relacionada con la transformada discreta de Fourier (del inglés *Discrete Fourier Transform* (DFT)), con la diferencia de que es una transformada real, debido a que los vectores base se componen exclusivamente de funciones coseno muestreadas [20].

Esta transformada ha tenido una gran aceptación dentro el tratamiento digital de imágenes y es por esto que la DCT se utiliza en los actuales estándares de compresión de imágenes. A continuación se presenta su definición y algunas de sus propiedades.

## C.1. DCT unidimensional (1D-DCT)

Formalmente, la DCT unidimensional es una función lineal invertible de $R^N$ en $R^N$ y podemos verla como una matriz cuadrada de NxN.

La DCT para un conjunto unidimensional de N muestras se puede expresar como:

$$C(u) = \alpha(u) \sum_{x=0}^{N-1} f(x) cos \left[ \frac{\pi(2x+1)u}{2N} \right], \tag{C.1}$$

para $u = 1, 2, \ldots, N-1$.

Similarmente, la transformada inversa está definida como:

$$f(x) = \sum_{u=0}^{N-1} \alpha(u)C(u)cos\left[\frac{\pi(2x+1)u}{2N}\right], \tag{C.2}$$

para $x = 1, 2, \ldots, N-1$.

En ambas ecuaciones C.1 y C.2, $\alpha(u)$ se define de la siguiente manera:

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} \text{ para} & u = 0 \\ \\ \sqrt{\frac{2}{N}} \text{ para} & u \neq 0 \end{cases} \tag{C.3}$$

## C.2. DCT bidimensional (2D-DCT)

La 2D-DCT es comunmente utilizada en imágenes y es una extensión directa de de la DCT unidimensional, está definida por la siguiente ecuación:

$$C(u,v) = \alpha(u)\alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y)cos\left[\frac{\pi(2x+1)u}{2N}\right] cos\left[\frac{\pi(2y+1)v}{2N}\right], \tag{C.4}$$

para $u, v = 1, 2, \ldots, N-1$.
La transformada inversa está definida como:

$$f(x,y) = \alpha(u)\alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y)cos\left[\frac{\pi(2x+1)u}{2N}\right] cos\left[\frac{\pi(2y+1)v}{2N}\right], \tag{C.5}$$

para $x, y = 1, 2, \ldots, N-1$. Para las ecuaciones C.4 y C.5, $\alpha(u)$ y $\alpha(v)$ están definidas en C.3.

## C.3. Propiedades de la DCT

Esta sección explica en términos generales algunas propiedades de la DCT [20], que tienen una importancia particular en aplicaciones de procesamiento de imágenes.

### C.3.1. Decorrelación

La principal ventaja de una operación de transformación en imagenes es la eliminación de redundancia entre píxeles vecinos. Podemos decir que la DCT exhibe buenas propiedades de decorrelación porque produce coeficientes decorrelados, que pueden ser codificados independientemente.

### C.3.2. Compactación de la energía

La eficacia de un esquema de transformación puede ser directamente medido por su habilidad de compactar los datos de entrada en el menor número de coeficientes posible. Esto permite al cuantificador descartar coeficientes con magnitudes pequeñas sin introducir distorción visual en la imagen reconstruida. La DCT presenta excelentes propiedades de compactación de energía para imagenes altamente correlacionadas.

### C.3.3. Separabilidad

La ecuación de la DCT C.4 puede ser expresada como:

$$C(u,v) = \alpha(u)\alpha(v) \sum_{x=0}^{N-1} cos \left[ \frac{\pi(2x+1)u}{2N} \right] \sum_{y=0}^{N-1} f(x,y) cos \left[ \frac{\pi(2y+1)v}{2N} \right], \tag{C.6}$$

para $u, v = 1, 2, \ldots, N-1$.

Esta propiedad nos permite calcular la 2D-DCT en dos partes: primero por columnas y luego por filas.

### C.3.4. Simetría

De la ecuación C.6 podemos observar que las operaciones que la conforman son funcionalmente idénticas. A este tipo de transformada se llama *transformada simétrica*. Una transformada separable y simétrica puede expresarse de la forma:

$$T = AfA \tag{C.7}$$

Donde $A$ es una matriz cuadrada de transformación de $N \times N$, cuyos elementos $a(i,j)$ para este caso particular están definidos como:

$$a(i,j) = \alpha(j) \sum_{j=0}^{N-1} cos \left[ \frac{\pi(2j+1)i}{2N} \right], \tag{C.8}$$

para $i, j = 0, 1, \ldots, N-1$. Donde $f$ es la matriz que representa la imagen de tamaño $N \times N$.

# Bibliografía


[1] S. Bhatt et al., "A personal mobile DRM manager for smartphones", *Computers & Security*, Vol. 28, No. 6, pp. 327-340, Sept. 2009.

[2] A. Al-Gindy et al., "A new watermarking scheme for colour images captured by mobile phone cameras", *International Journal of Computer Science and Network Security*, Vol.9, No.7, pp. 248-254, Jul. 2009.

[3] E. Kougianos et al., "Hardware assisted watermarking for multimedia", *Computers and Electrical Engineering*, Vol. 35, No. 2, pp. 339-358, Mar. 2009.

[4] A. Kejariwal et al., "Energy Efficient Watermarking on Mobile Devices Using Proxy-Based Partitioning", *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 14, No. 6, pp. 625-636, Jun. 2006.

[5] C. Rey and J.L. Dugelay, "A survey of watermarking algorithms for image authentication", *EURASIP Journal on Applied Signal Processing*, Vol. 2002, No. 6, pp. 613-621, Jan. 2002.

[6] A. Cheddad et al., "Digital Image Steganography: Survey and Analysis of Current Methods", in *Signal Processing*, Vol. 90, No. 3, pp. 727-752, Mar. 2010.

[7] I.J. Cox et al., "Watermarking applications and their properties", in *Proceedings of the The International Conference on Information Technology: Coding and Computing (ITCC'00)*, Las Vegas, NV, 2000, pp. 6-10.

[8] M. Arnold et al., *Techniques and Applications of Digital Watermarking and Content Protection*, MA:Artech House, 2003.

[9] S.P. Mohanty et al., "FPGA based implementation of an invisible-robust image watermarking encoder", in *Proceedings of the* $7_{th}$ *International Conference on Information Technology (CIT'04)*, Hyderabad, 2004, pp. 344353.

[10] A. Al-Gindy et al., "Enhanced dct based technique with shuffle scheme for robust image watermarking of handwritten signatures", in *International Conference on Communication and Power (ICCCP'07)*, Muscat, 2007, pp. 455-458.

[11] P. Sidiropoulos et al., "Invertible chaotic fragile watermarking for robust image authentication", *Chaos, Solitons & Fractals*, Vol. 42, No. 5, pp. 2667-2674, Dec. 2009.

[12] T. Lee and S.D. Lin, "Dual watermark for image tamper detection and recovery", *Pattern Recognition*, Vol. 41, No. 11, pp. 3497-3506, Nov. 2008.

[13] S.P. Mohanty, "A secure digital camera architecture for integrated real-time digital rights management", in *Journal Systems Architecture*, Vol. 55, No. 10-12, pp. 468-480, Oct. 2009.

[14] S. Walton, "Information authentification for a slippery new age", *Dr. Dobb's Journal*, Vol. 20, No. 4, pp. 1826, 1995.

[15] C.Y. Lin and S.F. Chang, "Semi-fragile watermarking for authenticating JPEG visual content", in *Proceedings of the SPIE International Conference on Security and Watermarking of Multimedia Contents II*, San Jose, Calif., 2000, pp. 140-151.

[16] C.T. Hsu and J.L. Wu, "Hidden digital watermarks in images", *IEEE Transactions on Image Processing*, Vol. 8, No. 1, pp. 5868, Jan. 1999.

[17] S.D. Lin and C.F. Chen, "A Robust DCT-Based Watermarking for Copyright Protection", *IEEE Transactions on Consumer Electronics*, Vol. 46, No. 3, pp. 415-421, Aug. 2000.

[18] S.S.Sherekar et al., "Role of Digital Watermark in e-governance and e-commerce", *International Journal of Computer Science and Network Security*, Vol. 8, No. 1, pp. 257-261, Jan. 2008.

[19] J. Fridrich et al., "Further attacks on Yeung-Mintzer fragile watermarking scheme", in *Proceedings of the SPIE International Conference on Security and Watermarking of Multimedia Contents II*, San Jose, Calif., 2000, pp. 428-437.

[20] S.A. Khayam. (2003). *The Discrete Cosine Transform (DCT): Theory and Application* [Online], Available FTP:davinci.tach.ula.ve Directory:vermig File: DCT_TR802.pdf

[21] K. Cabeen and P. Gent, *Image Compression and the Discrete Cosine Transform* [Online], Available FTP: online.redwoods.cc.ca.us Directory: instruct/darnold/LAPROJ/Fall98/PKen File: dct.pdf

[22] A.J. Arteaga. (2011). *Registro de desplazamiento con retroalimentación lineal* [Online], Available: http://samhain.softgot.com/criptografia/lecturasnotas.html

[23] P.C. Gil, *Avances en el estudio de la complejidad lineal del filtrado no lineal*, Dept. Ciencias y Tecnologías, Universidad de La Laguna, 1995.

[24] A. García, "Android", *Linux Magazine*, No. 49, pp. 50-53.

[25] J.A. Tudela, "Desarrollo de aplicaciones para dispositivos móviles sobre la plataforma de desarrollo Android de Google", Universidad Carlos III de Madrid Escuela Politécnica Superior, España, 2009.

[26] A.C. Alvarez. (2004). "La teoría del caos," in *Complejidad y caos: guía para la administración del siglo XXI* [Online]. Available: http://www.eumed.net/cursecon/libreria/2004/aca/aca.htm

[27] Darin McNabb Costa. *Peirce y la teoría del caos* [Online], Available FTP: unav.es Directory: gep File: JornadaArgentinaMcNabb.pdf

[28] C.H. Lai and C. Zhou, "Synchronization of chaotic maps by symmetric common noise", *Europhysics Letters*, Vol. 43, No. 4, pp. 376-380, 1998.