



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

RESUMEN DE LA MEMORIA
PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN
COMPUTACIÓN
ALFREDO COFRÉ E.
11/03/2007
PROF. GUÍA: SR. PATRICIO
INOSTROZA F.

PROCESAMIENTO DE FOTOGRAFÍAS DIGITALES EN DISPOSITIVOS
MÓVILES DE GAMA MEDIA

PROCESAMIENTO DE FOTOGRAFÍAS DIGITALES EN DISPOSITIVOS MÓVILES DE GAMA MEDIA

**MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN
COMPUTACIÓN**

ALFREDO ANDRÉS COFRÉ ESPINOZA

PROFESOR GUÍA:
PATRICIO INOSTROZA FAJARDIN

MIEMBROS DE LA COMISIÓN:
ALEJANDRO HEVIA ANGULO
RODRIGO PAREDES MORALEDA

SANTIAGO DE CHILE
MARZO DE 2007

Índice

1.- Introducción	1
1.1.- Motivación.....	1
1.2.- Objetivos	3
1.2.1.- Objetivo General	3
1.2.2.- Objetivos Específicos.....	3
1.3.- Sinopsis del Informe de Memoria	3
2.- Glosario	5
3.- Estado del Arte	6
3.1.- Investigación sobre los Dispositivos Objetivo	6
3.2.- Investigación sobre el Software Existente	9
3.2.1.- CameraFX	9
3.2.2.- CameraFX Pro.....	9
3.2.3.- FlexiCam.....	10
3.2.4.- FlexiCam Pro	10
3.2.5.- ImagePlus	10
3.2.6.- PhotoAcute Mobile	11
3.2.7.- Photo Editor	11
3.2.8.- PhotoRite	11
3.2.9.- PhotoRite SP.....	12
3.2.10.- Otros	12
3.2.11.- Sumario de la Investigación	12
3.3.- Investigación sobre Algoritmos de Procesamiento de Imágenes.....	14
3.3.1.- Operaciones Estadísticas	14
3.3.2.- Operaciones Espaciales	20
3.3.3.- Transformaciones geométricas, lógicas y aritméticas	23
3.4.- Sumario y Conclusiones de la Investigación.....	24
4.- Trabajo Realizado y Metodología	25
4.1.- Análisis Previo al Desarrollo del Software	25
4.1.1.- Consideraciones Matemáticas	25
4.1.2.- Consideraciones Técnicas	26
4.2.- Benchmarks	27
4.2.1.- Resultados de benchmarks	28
4.3.- Desarrollo	30
4.3.1.- Creación de un prototipo funcional	30
4.3.2.- Consideraciones de usabilidad	31
4.3.2.- Técnicas implementadas y optimizaciones.....	37
4.3.3.- Testing.....	38
4.4.- Inserción en el Mercado.....	45
5.- Resultados y Conclusiones.....	46
5.1.- Software Obtenido.....	46
5.2.- Desarrollos Futuros.....	46
6.- Bibliografía.....	48

1.- Introducción

El procesamiento digital de imágenes comprende, según [Escalera 2001], aquellos algoritmos cuya finalidad es conseguir una mejora en la apariencia de la imagen original. Esta mejora consiste en resaltar u ocultar determinadas características de la imagen. El tratamiento último es aplicado por un operador humano, resultando en una imagen más comprensible por el usuario final.

Existen en el país sobre diez millones de teléfonos celulares [SUBTEL 2005]. De éstos, la mayor parte de los teléfonos vendidos durante los años 2005-2006 corresponden a la llamada gama media, que agrupa, a grosso modo, a teléfonos con pantalla a color con precio inferior a los ochenta mil pesos (US\$149) en Junio del 2006. Aproximadamente la mitad de estos teléfonos de gama media son camera-phones, es decir, incorporan una cámara fotográfica.

Con el advenimiento de nuevos modelos de teléfonos celulares, paulatinamente los modelos de gama media se han ido dotando de mejores procesadores. Asimismo nuevos lenguajes de programación permiten, a costos comparativamente bajos, el desarrollo de aplicaciones y juegos que hasta hace poco sólo se encontraban en smartphones de gama alta. En particular, la llegada de la versión de Java para dispositivos móviles, J2ME [Sun 2005], ha ocasionado una avalancha de nuevas aplicaciones para teléfonos. Sin embargo, no existen muchas aplicaciones que permitan el mejoramiento de fotografías tomadas con camera-phone de forma móvil, usándose tradicionalmente para estos efectos el transferir las imágenes a un computador de escritorio y tratarlas posteriormente con un software como, por ejemplo, Picasa [Google 2006] o Photoshop. Se ha encontrado, además, que la mayoría de las aplicaciones móviles funcionan en una cantidad muy limitada de teléfonos de gama alta, puesto que han sido desarrollados para Symbian. Para la edición de imágenes en teléfonos de gama media, solo existe Photo Editor de Softex Digital [Softex 2006], desarrollado para teléfonos Benq-Siemens, el cual es un software muy básico, lo que en la práctica resulta poco útil si se intenta editar fotografías tomadas con el camera-phone.

El objetivo de esta memoria es crear un software de procesamiento de imágenes para teléfonos celulares que sea capaz de editar fotografías tomadas por camera-phones de gama media.

1.1.- Motivación

La motivación al elegir este tema como seminario de titulación es el que existe una amplia masa de usuarios de camera-phones que desaprovecha la capacidad fotográfica de éste, al no disponer de aplicaciones o utilidades que permitan mejorar la calidad de las fotografías obtenidas en terreno, debiendo esperar a disponer de un computador y algún medio de traspaso de archivos entre el computador y el teléfono para este fin. Si bien la óptica miniaturizada de las cámaras fotográficas de estos aparatos es limitada y propensa a sufrir de aberraciones ópticas, muchas de las fotografías que se desperdician pueden ser recuperadas mediante métodos de procesamiento de imágenes. La proposición de este proyecto es permitir procesar a las imágenes en forma temprana, in-situ, sin mediar un computador. Al efectuar esta

operación en el teléfono en vez de en el computador se obtiene una flexibilidad sin precedentes, permitiendo obtener muchas más imágenes de buena calidad en una misma cantidad de tomas comparando con un teléfono similar sin el software de procesamiento de imágenes. La lista de funcionalidades que se pretende incorporar a la aplicación es:

- Regulación de brillo.
- Regulación de contraste.
- Nitidez (sharpen).
- Suavizado (blur).
- Redimensionamiento (resize).
- Giro (rotate).
- Transformación a escala de grises (grayscale).
- Imagen en negativo (negative).
- Repujado (sharpen).

Quedan fuera del alcance de este proyecto las siguientes funcionalidades:

- Eliminación de ojos rojos.
- Detección de bordes.
- Detección de movimiento.
- Reconocimiento de patrones (ojos, caras, etc.).
- Aplicación de transformaciones geométricas distintas de las anteriormente señaladas.

Al investigar el abanico de aplicaciones similares a la que se propone construir, se encuentra que el mercado mundial ha reaccionado lentamente ante tal necesidad [Almalence 2006][Bitween 2006][Digia 2006][Nokia 2006][Softex 2006][WildPalm 2006][Yao 2006][Zenzis 2006], encontrándose nula oferta de procesamiento móvil de imágenes en el mercado nacional. Las aplicaciones que se encuentran a nivel mundial tienen portings muy limitados, existiendo versiones de estos softwares solamente para unos pocos teléfonos de gama alta.

Se ha detectado, asimismo, que las limitaciones técnicas que sufren los teléfonos actuales de gama baja y media son superadas rápidamente, observándose notorios incrementos en la potencia de las CPU, procesadores gráficos y capacidad de las cámaras durante el último año. Muchos teléfonos de precio inferior a \$80000 actualmente poseen poder de cálculo similar al de un PC Pentium III a 500 Mhz [ClubJava 2006]. Estas nuevas capacidades de cálculo permitirían a los teléfonos masivos actuales y futuros el realizar las funciones propuestas rápida y efectivamente.

Con respecto a posibles canales de venta y marketing, se ha considerado el uso de publishers de contenido para telefonía celular, enfatizando en establecer relaciones comerciales con operadoras de telefonía celular y portales web masivos con contenido wap. Estos canales permiten acceder al usuario final de manera expedita, al proveer de métodos probados de cobro y transferencia de contenido.

1.2.- Objetivos

1.2.1.- Objetivo General

El objetivo general de esta memoria es crear un software de edición digital de fotografías para dispositivos móviles de gama media.

1.2.2.- Objetivos Específicos

Los objetivos específicos de esta memoria son:

1. Realizar un estudio bibliográfico y técnico de la tecnología y softwares existentes.
2. Acotar y definir el conjunto de funciones de edición digital capaces de ser desarrolladas en un teléfono celular.
3. Implementar las funcionalidades antes definidas.
4. Desarrollar un prototipo con interfaces de usuario intuitivas y amigables.
5. Estudiar la inserción comercial de la aplicación.

1.3.- Sinopsis del Informe de Memoria

Este informe de memoria ha sido esquematizado siguiendo el desarrollo temporal de este proyecto de título. El resumen del contenido de los siguientes capítulos de este informe es el siguiente:

Capítulo 2 – Glosario: Compendio de términos técnicos usados en esta memoria.

Capítulo 3 – Estado del Arte: En este capítulo se efectúan tanto una revisión bibliográfica, sobre la web y en literatura impresa, como una investigación sobre el estado del arte en softwares de edición fotográfica para dispositivos móviles. Estas investigaciones se centran en tres ámbitos:

- **Investigación sobre dispositivos objetivo:** Se investiga sobre la oferta de camera-phones en el país, considerando tanto sus características técnicas como su precio, de modo de obtener una lista de teléfonos para los cuales portar el software a obtener de esta memoria.
- **Investigación sobre el software existente:** Investigando principalmente en la web, se obtiene una lista de softwares con algún grado de similaridad al software que se propone construir en esta memoria. Se exponen sus principales

características, fortalezas, precio y compatibilidad. Finalmente, se crea una tabla comparativa entre todos los softwares encontrados.

- **Investigación sobre algoritmos de procesamiento de imágenes:** Se exponen en este subcapítulo los principales algoritmos de procesamiento de imágenes, proporcionando ejemplos del uso de éstos sobre imágenes reales.

Capítulo 4 - Trabajo Realizado y Metodología: Se efectúan en este capítulo análisis sobre los algoritmos de procesamiento de imágenes y métodos de implementación considerando las limitantes técnicas de los teléfonos celulares. Se expone en detalle el desarrollo de programas de benchmarks y de un prototipo funcional de la aplicación.

Capítulo 5 - Resultados y Conclusiones: Se exponen y comentan los resultados obtenidos del desarrollo de esta memoria, estudiando métodos para una posible inserción en el mercado.

Capítulo 6 - Bibliografía: Listado de fuentes y referencias consultadas durante el desarrollo de este proyecto.

2.- Glosario

API: Set de clases que provee de funcionalidades extraordinarias a J2ME. Para poder trabajar sobre un teléfono celular, la API debe ser soportada por éste.

Benchmark: Prueba de rendimiento. Usualmente se usan para evaluar y comparar las características del dispositivo o aplicación objetivo.

Camera-Phone: Teléfono con cámara fotográfica incorporada.

Gama: Segmentación artificial del espectro de teléfonos existentes en el mercado. Aunque no es estándar y varía a medida que cambia el espectro de modelos y capacidades disponibles y depende, además, de la visión del mercado que se tenga, es una expresión muy usada en la industria de la telefonía móvil. Generalmente se habla de teléfonos de gama baja (tecnología TDMA, pantallas monocromáticas), gama media (teléfonos de tercera generación, pantalla a color, posiblemente con cámara fotográfica) y gama alta (teléfonos de alto costo usualmente con características especiales, tales como reproductor de MP3, grabación de video, pantallas de gran tamaño, teclado especial, smartphones, etc.). En términos de costo, se habla de teléfonos de prepago de valor inferior a \$40.000 en la gama baja, de valor entre \$40.000 y \$80.000 para la gama media, y de costo superior a \$80.000 para la gama alta.

Es importante observar que la mayor parte de los teléfonos de prepago que se venden en el país corresponden a la gama media.

Jar: Paquete comprimido que contiene a la aplicación Java y a sus recursos internos.

J2ME: Versión para dispositivos móviles de Java. Existe en versiones MIDP1 y MIDP2; asimismo, MIDP2 se divide en CLDC 1.0 y CLDC 1.1. Esta clasificación hace referencia a la capacidad de procesamiento del dispositivo móvil. MIDP2 dispone de una API más amplia que MIDP1 que permite muchas más funcionalidades, a costa de requerir procesadores más potentes.

Operadora: Empresa proveedora de servicio de telefonía celular.

Porting: Proceso de modificar un software creado para cierto modelo de teléfono para que funcione en otro modelo o marca. Este proceso puede ser algo tan trivial como modificar el tamaño de la pantalla o tan complicado como, prácticamente, rescribir la aplicación.

Smartphone: Teléfono que incorpora funcionalidades de PDA, tales como suites de ofimática. Esta definición incluye a las PDA con funcionalidad de teléfono.

Symbian: Sistema operativo para teléfonos celulares, desarrollado por Symbian Corporation, propiedad de Nokia, Sony Ericsson, Samsung y otros fabricantes de celulares. Permite el uso de programas creados en C++, mediante el uso de bibliotecas especiales. Es usado por teléfonos de gama media/alta, tales como, por ejemplo, Sony Ericsson W600/W800 y Nokia Serie 60 y 80.

3.- Estado del Arte

La investigación efectuada durante esta memoria se ha enfocado principalmente en tres puntos:

1. Investigación del mercado de teléfonos celulares y de las capacidades de los dispositivos móviles.
2. Investigación del mercado del software de edición fotográfica móvil.
3. Investigación sobre algoritmos usados en el procesamiento de imágenes.

3.1.- Investigación sobre los Dispositivos Objetivo

Dado que se desea que el software que se propone construir corra sobre camera-phones existentes en el mercado chileno, ha sido natural estudiar la oferta existente en los sitios web de los principales operadores nacionales, Movistar [Movistar 2006], ENTEL PCS [ENTEL 2006] y Claro Chile [Claro 2006].

Durante esta investigación se propuso determinar qué teléfonos, presentes en la oferta nacional, cumplen tanto con características técnicas compatibles con el software propuesto como con precios de venta tales que se les pudiese clasificar como teléfonos de gama media. Las características técnicas de los teléfonos se obtuvieron de una serie de fuentes, entre las que se destacan GSM Arena [GSMarena 2006], Phone Scoop [PhoneScoop 2006] y Softpedia [Softpedia 2006].

De estas fuentes se ha obtenido la Tabla 1, que define los teléfonos que caen dentro de la categoría "gama media", es decir, que en Junio de 2006 cuesten menos de \$80.000 en modalidad de prepago, y que sean compatibles con la especificación MIDP2. Es necesario acotar que existen varios modelos en la oferta nacional de los cuales no se pudo obtener o confirmar especificaciones técnicas (principalmente, teléfonos LG, Movistar, Samsung y Motorola), por lo que no se consideraron en esta lista.

De esta lista se obtuvo, en Noviembre de 2006, sólo un teléfono que cumpliera con las características deseadas, es decir, que calificase en cuanto a precio, disponibilidad y rendimiento y considerando, además, las APIs que es necesario utilizar (JSR-135 MMAPI y JSR-75 Fileconnection - PIM API). Este teléfono es el Motorola L6, disponible como prepago a \$79.900 en ENTEL, Claro y Movistar.

Dada el reducido número de teléfonos disponibles en la actualidad que cumplen con las características técnicas deseadas, se ha desarrollado la Tabla 2, que contiene teléfonos que, siendo técnicamente candidatos aceptables, no caen exactamente dentro del criterio económico de selección. Estos modelos en particular se acercan notablemente al precio establecido como límite, por lo que estimamos que en poco tiempo estos modelos deberían caer dentro de los criterios de selección.

La masividad de los teléfonos fue obtenida gracias a la gentil colaboración de AndinaTech [AndinaTech 2006], empresa dedicada a la publicación de videojuegos y contenido para teléfonos celulares.

Tabla 1: Modelos de teléfonos candidatos (*)()**

Marca	Modelo	Precio	Operadora
Alcatel	551	\$ 44.900	Movistar
Alcatel	557	\$ 39.900	ENTEL PCS
Alcatel	757	\$ 69.900	Movistar
Alcatel	C552	\$ 44.900	ENTEL PCS
Alcatel	C651	\$ 44.900	ENTEL PCS
Alcatel	C652	\$ 57.900	Movistar, ENTEL PCS
Motorola	C650	\$ 39.900	Movistar, ENTEL PCS
Motorola	L6	\$ 79.900	Movistar, ENTEL PCS, Claro
Nokia	3220	\$ 49.900	Movistar, ENTEL PCS, Claro
Nokia	5140	\$ 65.900	Movistar
Nokia	6020	\$ 64.900	Movistar, ENTEL PCS
Nokia	6030	\$ 41.900	Movistar
Nokia	6070	\$ 69.900	ENTEL PCS
Sagem	X5-2	\$ 39.900	ENTEL PCS
Sagem	X7	\$ 49.900	ENTEL PCS
Samsung	X620	\$ 59.900	ENTEL PCS
Siemens	C65	\$ 62.900	ENTEL PCS
Siemens	C66	\$ 49.900	Movistar, ENTEL PCS
Siemens	CF75, CF76	\$ 67.900	Movistar, ENTEL PCS
Siemens	CX65	\$ 64.900	ENTEL PCS
Siemens	M65, M75	\$ 59.900	Movistar, ENTEL PCS
Siemens	S66	\$ 69.900	Movistar
Sony-Ericsson	K300, K310	\$ 69.900	Movistar, ENTEL PCS

Tabla 2: Modelos de teléfonos que podrían seleccionarse en el futuro(*)

Marca	Modelo	Precio	Operadora
Motorola	V555	\$ 89.900	Movistar
Nokia	6101	\$ 89.900	Movistar
Sony-Ericsson	K510	\$ 94.900	Movistar, ENTEL PCS, Claro

(*): La selección podría variar dependiendo de la disponibilidad de tales modelos, de la aparición de nuevos modelos de gama media y de la evolución de los precios de los teléfonos.

(**): A fecha de 22 de Noviembre de 2006, varios de los teléfonos de la Tabla 1 ya no están disponibles para venta directa e incluso algunos han sido objeto de reemplazo por nuevos modelos. Así, por ejemplo, el teléfono Siemens C65 ha sido reemplazado por el Benq-Siemens C72, disponible en ENTEL PCS a \$44.900.

Investigando la capacidad de cálculo de los teléfonos seleccionados, se ha obtenido que un teléfono electrónicamente similar al Motorola SLVR L6, el antiguo modelo Motorola RAZR V3, obtiene los siguientes resultados de desempeño en las pruebas de ClubJava [ClubJava 2006]:

- Velocidad de procesamiento de la Máquina Virtual de Java: 3,6 Mhz.
- Velocidad de procesamiento equivalente a un PIII de 66,1 Mhz.
- Velocidad de lectura de memoria: 2334 Kb/s.
- Velocidad de escritura a memoria: 2346 Kb/s.
- Velocidad de copia en memoria: 6021 Kb/s.

3.2.- Investigación sobre el Software Existente

Tal como se mencionó en la introducción, el software de edición fotográfica para dispositivos móviles existente a nivel mundial se concentra principalmente o en teléfonos de gama media-alta o en modelos muy específicos de algunos proveedores como Benq-Siemens [BenqSiemens 2006].

Los principales softwares encontrados al efectuar una exhaustiva revisión bibliográfica en la web se detallan a continuación. En la página 13 se encuentra la tabla 3, donde se muestra un resumen con las principales características de éstos.

3.2.1.- CameraFX

Desarrollado por WildPalm [WildPalm 2006], este software está portado para Nokia 6600 y Siemens SX-1. Permite efectuar:

- Zoom digital.
- Estiramientos horizontales o verticales.
- Colores en negativo o escala de grises.
- Manejo de brillo y contraste.
- Ajustes de rojo, verde y azul.
- Intercambio de colores.
- Giros con ángulo libre.
- Flip horizontal o vertical.
- Agregar marcos de fotos.
- Efectos de proyección sobre primitivas 3D, como cubos; clonación; etc.

Este software cuesta €9.99 (\$ 6877) a la fecha del 4 de Julio de 2006.

3.2.2.- CameraFX Pro

Evolución de CameraFX, este software está además portado para Nokia N90. A las características de CameraFX se agrega:

- Modo de color "Highlight".
- Perfiles de color (sepia, B/N, etc.).
- Clonación en espejo.
- Aplicación de transformaciones tipo "Ojo de pez", detección de bordes, remolino, caleidoscopio.
- Repetición múltiple, mosaico.
- Proyecciones sobre cilindros, prismas, túnel, perspectivas y laberintos.

Su precio es de €9.99 (\$ 6877) al 4 de Julio de 2006.

3.2.3.- FlexiCam

FlexiCam, o Flexible Camera, es un software freeware desarrollado por Yao [Yao 2006]. Los principales efectos encontrados acá son:

- Proyecciones cilíndricas.
- Suavizado (blur).
- Realce de nitidez (sharping).
- Modificación de la luminosidad.
- Modificación de color (RGB).
- Animaciones.
- Aplicación de mascararas. Las mascararas pueden ser creadas por el usuario, en formato bmp.
- Filtros.

Es compatible con los siguientes dispositivos: Nokia 3600, 6600, 6650, 7650, Siemens SX-1. Sin embargo, debiera poder correr en cualquier dispositivo Nokia Serie 60.

3.2.4.- FlexiCam Pro

Este software es la versión comercial de FlexiCam. Posee la misma compatibilidad que el anterior software, agregando gran cantidad de nuevos filtros y opciones.

Registrar este software cuesta, al día 4 de Julio de 2006, US\$10 (\$5377) o €8 (\$5507).

3.2.5.- ImagePlus

ImagePlus es un editor desarrollado por Digia [Digia 2006] para teléfonos Nokia Serie 60. Sus principales características son:

- Inserción de textos y anotaciones.
- Creación de presentaciones.
- Efectos de suavizado (blur) y distorsión.
- Rotación en ángulo libre, zoom, escalado, reflexión y desplazamiento.
- Unión de imágenes usando transparencia.
- Efectos sepia.
- Manejo de brillo y contraste.
- Integración con la aplicación por defecto de la cámara del teléfono S60.

Su precio, al 4 de Julio de 2006, es de €9.95 (\$6850).

3.2.6.- PhotoAcute Mobile

Este software, desarrollado por Almalence Inc. [Almalence 2006], es parte de la suite PhotoAcute. Su objetivo es producir imágenes de resolución mayor a la original, utilizando avanzadas funciones de generación de nuevos píxeles. Se encuentra disponible para Nokia Serie 60 (Nokia 3650, 3660, 6600, 6620, 6260, 6630, 6680, 6681, 6682, N70 y N90), y Sony-Ericsson UIQ (P800, P900 y P910).

Su precio es de US\$ 16 (\$8603) al 4 de Julio de 2006.

3.2.7.- Photo Editor

Desarrollado por Softex Digital [Softex 2006], Photo Editor es un editor de imágenes que permite:

- Ajustar balance de color, brillo y contraste.
- Fragmentar, rotar y redimensionar.
- Deformar.
- Agregar elementos divertidos, como íconos y marcos de fotos.
- Aplicar filtros sepia, solarización, repujado (emboss) y otros.
- Aplicar texto.

Este software ha sido desarrollado para ser distribuido de forma embebida en algunos modelos de teléfonos Siemens.

Lamentablemente, no existe una lista oficial de teléfonos que corran este software; sin embargo, el autor lo ha visto incorporado en teléfonos Benq-Siemens de gama baja, como el Siemens A-76.

3.2.8.- PhotoRite

Zenzis Limited [Zenzis 2006] ha creado una completa serie de softwares de fotografía y edición digital. En el ámbito móvil, se encuentra PhotoRite y PhotoRite SP.

PhotoRite es un poderoso software, desarrollado para teléfonos UIQ Sony-Ericsson P800 y P900. Entre sus características se encuentran:

- Permite tomar fotografías desde el interior del software para su inmediato postproceso.
- Rápida generación de thumbnails.
- Auto corrección de imágenes.
- Efectos de lentes y filtros.
- Efectos sepia y blanco y negro.

- Marcos de imágenes.
- Deshacer paso a paso ilimitado.
- Manejo de contraste y brillo.
- Corrección de errores en la exposición.
- Balance de colores.
- Deformaciones.

El costo de esta aplicación al 4 de Julio de 2006 es de US\$9.99 (\$5371).

3.2.9.- PhotoRite SP

Este software es el equivalente de PhotoRite, para teléfonos Nokia Serie 60. Los dispositivos oficialmente soportados son el Nokia N70, N90, 3230, 6600/6620, 6260, 6630, 6670, 7610 y 6680. Posee un control más preciso de la cámara integrada que PhotoRite, permitiendo además el obtener imágenes en distintas resoluciones. Soporta todas las características de PhotoRite, agregando la posibilidad de hacer zoom digital de hasta 8x.

El precio de este software, al 4 de Julio de 2006, es de US\$ 19.95 (\$10727).

3.2.10.- Otros

Se han encontrado otros softwares dentro del mercado que, si bien efectúan edición digital de fotografías, poseen características demasiado básicas u orientadas completamente al entretenimiento, realizando deformación de caras o agregando a las fotografías elementos tales como peinados, sombreros, narices de payaso, etc. Estos son:

- CrazyWarp (UIQ, S60), <http://www.crazywarp.com>
- FaceWarping (S60), <http://www.cognimatics.com>
- FotoEdit (S60), <http://infospacegames.com>
- HairStyle Mobile (S60), <http://hairstyle-mobile.softonic.de>

3.2.11.- Sumario de la Investigación

Tal como se observa en la tabla 3 (página siguiente), el espectro de softwares existentes se concentra de manera casi exclusiva en teléfonos de gama alta, tales como los Nokia Serie 60, Sony Ericsson UIQ y Siemens SX-1. El espectro de funciones varía desde softwares limitados a una única función, como PhotoAcute Mobile, hasta softwares muy completos, tal como Camera FX Pro.

El espectro de precios varía desde \$5.371, para PhotoRite, hasta \$10.727 para PhotoRite SP; este hecho muestra que el porting es un factor fundamental al decidir el precio de venta de la aplicación.

Es importante notar que existe un software bastante completo, FlexiCam, que es freeware; asimismo PhotoAcute Mobile, que tan sólo redimensiona imágenes es el segundo software más caro, costando \$8.603.

Tabla 3: Software existente en el mercado v/s características principales

Características\ Software	CameraFX	Camera FX Pro	FlexiCam	FlexiCam Pro	ImagePlus	PhotoAcute Mobile	Photo Editor	PhotoRite	PhotoRite SP	PhotoLab*
Precio (Pesos chilenos)	\$ 6.877	\$ 6.877	Freeware	\$ 5.377	\$ 6.850	\$ 8.603	Embebido	\$ 5.371	\$ 10.727	A definir**
Compatibilidad	Nokia S60, Siemens SX-1	Nokia S80	Nokia S60, Siemens SX-1	Nokia S60, Siemens SX-1	Nokia S60	Nokia S60, Sony-Ericsson UIQ	Siemens	Sony-Ericsson UIQ	Nokia S60	Motorola L6***
Ajuste de brillo	X	X	X	X	X		X			X
Ajuste de colores RGB	X	X	X	X			X	X	X	
Ajuste de contraste	X	X			X		X			X
Animaciones, presentaciones			X	X	X					
Auto corrección								X	X	
Clonaciones, mosaicos		X								
Colores en negativo	X	X								X
Detección de bordes		X								
Escala de Grises	X	X						X	X	X
Filtros			X	X			X	X	X	X
Intercambio de colores	X	X								
Marcos de fotos	X	X					X	X	X	
Máscaras			X	X			X			
Proyecciones	X	X	X	X						
Realce de nitidez			X	X						X
Recorte, fragmentación					X		X			
Redimensionamientos	X	X			X	X	X			X
Reflexiones	X	X			X					
Rotación en ángulo libre	X	X			X		X			X
Sepia		X			X			X	X	
Suavizado			X	X	X					X
Textos					X		X			
Thumbnails								X	X	
Transformaciones, distorsiones		X			X		X	X	X	
Unión de imágenes					X					
Zoom	X	X			X				X	

* PhotoLab: software que se propone construir en esta memoria

** El precio deberá ser definido al negociar una eventual salida al mercado.

*** Compatibilidad preliminar.

3.3.- Investigación sobre Algoritmos de Procesamiento de Imágenes

Los algoritmos usados en el procesamiento de imágenes se agrupan en tres familias principales, capaces de proveer, por sí solas o usándose en conjunto, las funcionalidades deseadas.

La primera familia corresponde a algoritmos independientes de la posición de los píxeles, llamados algoritmos sobre **operaciones estadísticas**. En esta familia se modifican directamente los valores de los píxeles de la imagen operando, principalmente, a través del histograma de la imagen.

La segunda familia corresponde a algoritmos dependientes de la posición del píxel, llamados algoritmos sobre **operaciones espaciales**. Estos algoritmos se basan principalmente en los operadores de convolución y correlación.

La tercera familia corresponde a algoritmos basados en el muestreo de frecuencias, llamados algoritmos sobre el **dominio de las frecuencias**. En esta familia se modifica la transformada de Fourier de la imagen. No usaremos en esta memoria algoritmos de esta familia.

Es necesario considerar además los algoritmos que aplican **transformaciones aritméticas o lógicas** que permiten, por ejemplo, mezclar imágenes y máscaras. Existen además las **transformaciones geométricas**, que modifican las relaciones espaciales entre los píxeles.

3.3.1.- Operaciones Estadísticas

En el rango de operaciones estadísticas destacan nítidamente los algoritmos de procesamiento del histograma de intensidad de la imagen. Este tipo de algoritmos permite efectuar de manera bastante barata, en términos de procesamiento y memoria, modificaciones al mapeo de las intensidades de los píxeles, es decir, a su luminosidad. Estas técnicas se denominan **transformaciones de intensidad** [Mai 2006] .

La transformación de intensidad es un proceso que convierte a un píxel original en otro nuevo, basado en alguna función predefinida. Estas transformaciones se aplican sobre tablas de consulta, que son estructuras de datos que asocian a cada intensidad antigua el nuevo valor de la intensidad. Luego, al construir la nueva imagen, se reemplazan los valores de las intensidades antiguas por las nuevas. El tamaño de la tabla de consulta es de 2^n , donde n es la profundidad de bits de la paleta de intensidad de colores. Típicamente, para una paleta de 24 bits de colores, se tienen 256 niveles de intensidad para cada canal RGB o HSI, es decir, $n = 8$.

Las operaciones que se pueden efectuar con esta técnica, variando la función de transformación, son principalmente

- **Correcciones al factor gamma:** Son correcciones no lineales al brillo de la imagen. Sirven para modificar el brillo de una imagen o modificar el rendering de la imagen en pantalla. En general, el factor de corrección gamma es:

$$\text{entrada} = \text{salida}^{1/\gamma}$$

Si $0 < \gamma < 1$, γ crea curvas de brillo exponenciales que oscurecen la imagen. Si $\gamma > 1$, el resultado son curvas logarítmicas que abrillantan la imagen.

Típicamente, los monitores CRT tienen valores de γ de entre 1.4 y 2.8. Este valor no está documentado para pantallas de celular. En la Imagen 1 se puede apreciar un ejemplo de corrección gamma con (a) Curva de $\gamma = 0,45$; (b) Imagen corregida con $\gamma = 0,45$; (c) Curva de $\gamma = 2,2$; (c) Imagen corregida con $\gamma = 2,2$.

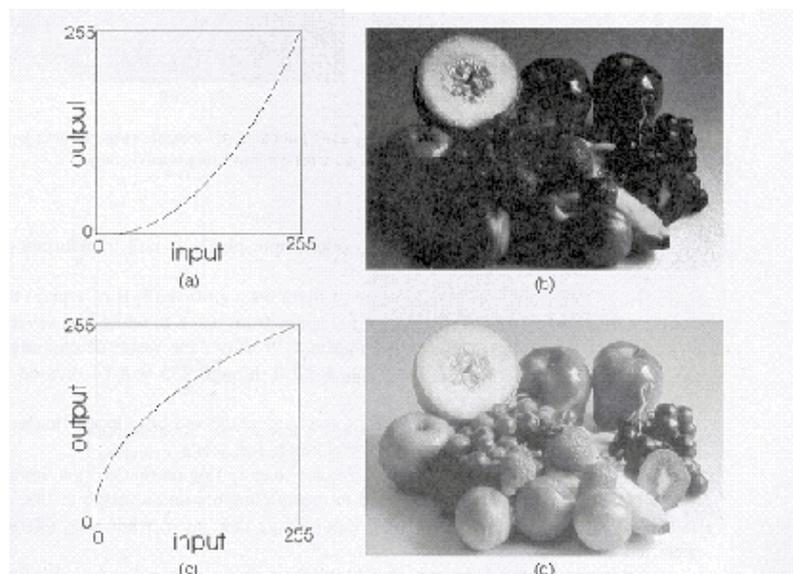


Imagen 1: Correcciones Gamma

- **Manejo del contraste:** El contraste de una imagen es la distribución entre píxeles claros y oscuros. Las imágenes en escala de grises de bajo contraste son predominantemente oscuras, si los píxeles se agrupan a la izquierda del histograma; claras, si se concentran a la derecha, o grises, si se concentran en el centro. En estos casos, se está usando una pequeña muestra de intensidades de entre toda la paleta de intensidades disponible.

La Imagen 2 es un ejemplo de una imagen de bajo contraste y predominantemente gris. El histograma de esta imagen fue obtenido usando Picasa2 [Google 2006]).



Imagen 2: Imagen con contraste bajo y su histograma

Las imágenes con alto contraste tienen tanto zonas muy claras como muy oscuras, pero en regiones del histograma excesivamente separadas entre sí. Los histogramas de imágenes con alto contraste tienden a tener dos máximos, uno ubicado en la región clara y otro en la región oscura (Imagen 3).

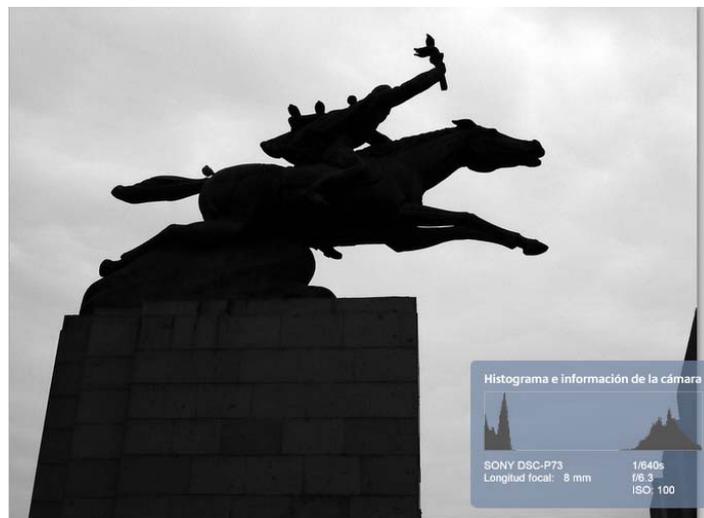


Imagen 3: Imagen con alto contraste y su histograma

Las imágenes con buen contraste muestran un histograma en el que las intensidades se reparten uniformemente a lo largo de la abscisa. No existen valles o montes importantes (Imagen 4).

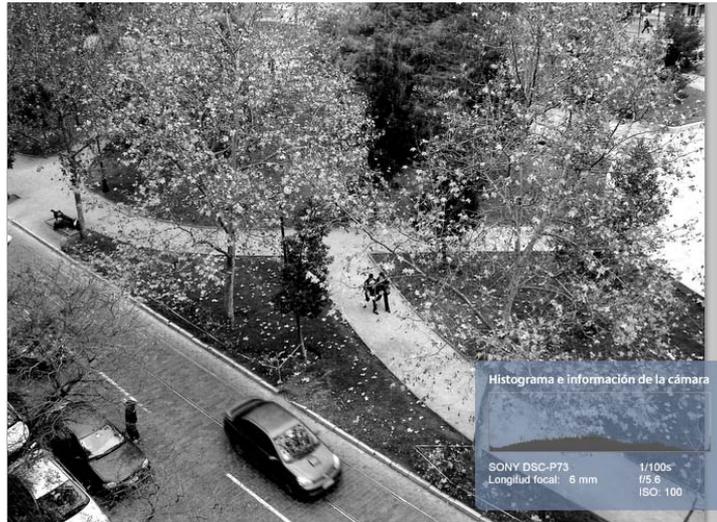


Imagen 4: Imagen con buen contraste y su histograma

Se aplica *estiramiento del contraste* cuando el rango dinámico de la imagen es muy angosto; esto es, cuando las imágenes tienen bajo contraste. Funciona mejor cuando las imágenes tienen una distribución aproximadamente Gaussiana de intensidades (histograma Imagen 2). Las técnicas más comunes de estiramiento de contraste son el estiramiento de contraste básico y el end-in search.

En el caso en que los píxeles se agrupan en el una parte del histograma es mejor aplicar estiramiento básico. Esta técnica expande el histograma de modo de cubra todo el rango de píxeles (Imagen 5). Se usan el valor más alto y más bajo de los píxeles en la transformación. La ecuación es:

Ecuación 1: Estiramiento básico de contraste

$$\text{nuevo_pixel} = \frac{\text{pixel_antiguo} - \text{bajo}}{\text{alto} - \text{bajo}} * 255$$

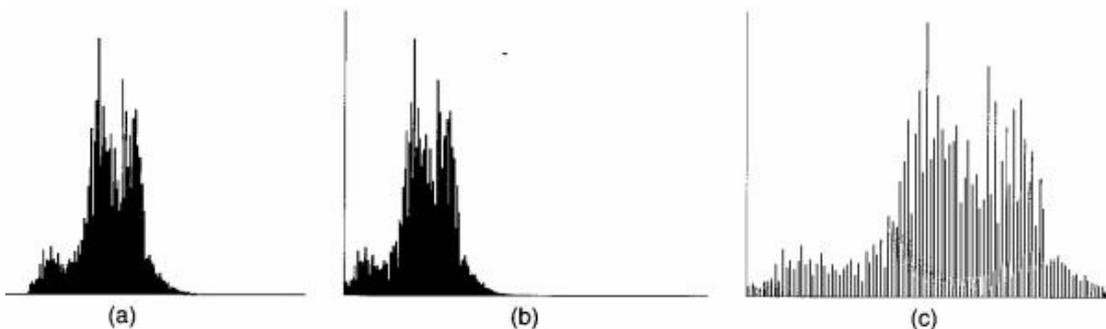


Imagen 5: (a) Imagen original; (b) histograma bajado; (c) (alto-bajo)*255/(alto-bajo)

El píxel del valor más bajo ahora es 0, mientras que el píxel con valor más alto es 255.

El segundo método es end-in search. Funciona bien con imágenes que tienen un amplio rango dinámico pero concentran sus píxeles en una parte del histograma. Sin embargo, usa intensivamente el procesador, por lo que no se usará en esta memoria.

Cuando se tiene una imagen de alto contraste o con detalles finos en las zonas oscuras, es útil aplicar la *ecualización de histograma*. En esta técnica, se redistribuye la distribución de intensidades. Si el histograma de una imagen tiene picos y valles, el histograma modificado aún los poseerá, pero posiciones distintas a las originales. En términos intuitivos esta técnica, más que “aplanar” el histograma, lo “expande” sobre todo rango de valores posibles. Esto se puede apreciar claramente en la Imagen 6.

El algoritmo funciona así:

- Se computa el histograma.
- Se calcula la suma normalizada del histograma.
- Se transforma la imagen de entrada en una imagen de salida.

La suma normalizada se calcula con un arreglo de tamaño 2^n ($n = 8$ típicamente; es decir, se tienen colores de 8 bits). En este arreglo, el elemento 1 contiene la suma de los elementos del histograma 1 y 0. El elemento 256 contiene la suma de los elementos del histograma 255, 254, 253, ... , 1, 0. Luego, este arreglo se normaliza multiplicando cada elemento por (valor máximo de píxel/cantidad de píxeles). Para una imagen de 8 bits de color de 512X512 píxeles, esta constante debería ser 255/262144.

Como resultado del segundo paso se obtiene una tabla de consulta que, operando sobre la imagen original, entrega una imagen ecualizada.

El resultado de aplicar la ecualización de histograma se puede apreciar en la Imagen 6.



Imagen 6: (Izquierda) Imagen original; (derecha) Imagen ecualizada

Es importante acotar que si bien estas operaciones están definidas para imágenes en escala de grises, se puede operar sobre imágenes a color aplicando cada operación sobre cada canal RGB por separado.

- **Manejo del brillo:** El brillo se puede variar comprimiendo el histograma hacia la derecha o la izquierda, dependiendo de si se desea aclarar u oscurecer la imagen.

El uso de las transformaciones de intensidad exige, claramente, poder trabajar sobre los valores de las intensidades de los píxeles. Sin embargo, los colores se encuentran definidos, en las imágenes, según el modelo RGB, por lo que es necesario transformar los colores al modelo HSI, para operar luego sobre el componente I. Para obtener los valores HSI desde el modelo RGB, se aplican las siguientes transformaciones:

Ecuación 2: Transformaciones RGB-HSI

$$I = \frac{R + G + B}{3}$$

$$H = \arctan\left(\frac{\sqrt{3}(G - B)}{(R - G) + (R - B)}\right)$$

$$S = 1 - \frac{\min(R, G, B)}{I}$$

Para obtener la transformación inversa, se define que si $0^\circ < H \leq 120^\circ$, $H = H$; si $120^\circ < H \leq 240^\circ$, $H = H - 120^\circ$, y si $240^\circ < H \leq 360^\circ$, $H = H - 240^\circ$. Entonces,

Ecuación 3: Transformaciones HSI-RGB

$$B = \frac{1}{3}(1 - S)$$

$$R = \frac{1 + S \cos H}{3 \cos(60 - H)}$$

$$G = 1 - (B + R)$$

Otras operaciones sobre el dominio del espacio incluyen variaciones en la distribución de rojo, verde y azul y el reemplazo de colores. Estas operaciones se pueden efectuar sobre el modelo RGB.

3.3.2.- Operaciones Espaciales

Las operaciones espaciales son operaciones espacialmente dependientes, es decir, la transformación depende de las coordenadas de la imagen sobre la que se aplica. Estas operaciones no preservan la forma del histograma de intensidades.

Parte fundamental de esta clase de operaciones y del procesamiento de imágenes en sí son los filtros de convoluciones [Escalera 2001][Mai 2006] [Patin 2003][Richter 1999]. Estos permiten implementar gran cantidad de efectos, tales como suavizado (blur), mejorar la nitidez (sharpen), aumentar o disminuir características de la imagen, etc. Se componen de una ventana deslizante T , la que se aplica al píxel $I(X,Y)$ de la imagen I , como:

Ecuación 4: Función de Convolución

$$T * I(X,Y) = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} T(i,j)I(X-i,Y-j) ,$$

donde n es el ancho de la ventana de convolución y m es el alto de la misma. Las ventanas de convolución son generalmente matrices de entre 2X2 y 5X5 valores, usándose comúnmente ventanas de 3X3 [Mai 2006] Variando los valores de la ventana T se pueden obtener, por ejemplo, los siguientes efectos:

- **Mejoramiento de la nitidez (Sharpen):** Muy útil para hacer evidentes detalles pequeños de la imagen. Típicamente se usan para esto matrices similares a $\begin{bmatrix} -1 & -1 & -1 \\ -1 & 16 & -1 \\ -1 & -1 & -1 \end{bmatrix}$.



Imagen 7: Mejoramiento de la nitidez (sharpen); (izquierda) Imagen original; (derecha) Imagen con nitidez aumentada

- **Suavizado (Blur):** Suaviza detalles de la imagen, "promediando" píxeles vecinos y, asimismo, mejorando la escalabilidad. Se suelen usar matrices de la forma $\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$ (gaussian blur).



Imagen 8: Suavizado (blur); (izquierda) Imagen original; (derecha) Imagen suavizada

- **Emboss (Repujado):** Es un filtro de detección de bordes. Se usan matrices del tipo $\begin{bmatrix} -2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -2 \end{bmatrix}$. Suele aplicarse, antes de emboss, un filtro de color o a blanco y negro.



Imagen 9: Emboss; (izquierda) Imagen original; (derecha) Imagen tratada

- **Median filtering:** Permite eliminar ruido de la imagen, preservando los bordes de los objetos de una manera mejor que aplicando blur. Se usa una ventana (template) sobre la imagen. Los valores de los píxeles en el template son guardados, y su mediana (el valor del medio en una lista ordenada de valores, o el promedio entre los valores medianos si el template es de tamaño par) se despliega en la imagen de salida. El template suele ser de entre 3X3 a 7X7 píxeles.



Imagen 10: Median Filtering; (izquierda) Imagen original; (derecha) Imagen filtrada

- **Remoción de graduaciones:** Permite remover sombras graduales. Instintivamente, una sombra gradual es una sombra fuerte que va gradualmente perdiendo fuerza a medida que se recorre la imagen. Esta técnica permite obtener una imagen con iluminación más homogénea en todos sus sectores.

La remoción de graduaciones se realiza subdividiendo la imagen en rectángulos, evaluando la intensidad promedio tanto en cada rectángulo como en la imagen completa. Luego, se suma o resta a cada píxel una constante elegida de modo tal que todos los rectángulos de la imagen tengan un promedio igual.



Imagen 11: Remoción de graduación; (izquierda) Imagen original; (derecha) Imagen tratada

3.3.3.- Transformaciones geométricas, lógicas y aritméticas

- **Aplicación de máscaras:** Permite aplicar marcos o textos a las imágenes, reemplazando zonas seleccionadas de la imagen original por píxeles de otra imagen. Se efectúa usando como máscara una imagen con un color "transparente", es decir, si un píxel de la máscara tiene color transparente se conserva el píxel de la imagen original, y en caso contrario, el píxel de la máscara, y efectuando entre ambas una operación lógica AND.
- **Rotación:** supondremos que deseamos rotar una imagen con respecto a su centro en un ángulo libre α . En este caso, la ecuación es:

Ecuación 5: Rotación en ángulo libre

$$\begin{bmatrix} x_f \\ y_f \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \frac{x_{\max}}{2} \\ 0 & 1 & \frac{y_{\max}}{2} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \alpha & -\operatorname{sen} \alpha & 0 \\ \operatorname{sen} \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -\frac{x_{\max}}{2} \\ 0 & 1 & -\frac{y_{\max}}{2} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

- **Escalamiento:** Asimismo, supondremos que escalaremos la imagen en un factor similar tanto en el eje vertical como en el horizontal. En este caso, usando coordenadas homogéneas, la ecuación de escalamiento es:

Ecuación 6: Escalamiento

$$\begin{bmatrix} x_f \\ y_f \\ 1 \end{bmatrix} = \begin{bmatrix} 1/a & 0 & 0 \\ 0 & 1/a & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix},$$

donde (x_i, y_i) es el píxel inicial, (x_f, y_f) es el píxel final, y a es el factor de escalamiento.

3.4.- Sumario y Conclusiones de la Investigación

La investigación con respecto a los distintos softwares existentes en el mercado se resume, considerando sus características principales, en la Tabla 3.

De esta tabla se puede obtener que, si bien las características ofrecidas por los softwares es adecuada a las necesidades de la edición fotográfica in-situ, el espectro de teléfonos en los que se puede efectuar tal operación es muy estrecho, limitándose a modelos Nokia S60 y S80, Sony-Ericsson UIQ y Siemens SX-1, todos de gama alta, más la solución embebida de Siemens. Es por esto que la elección de construir un software de similares características en Java es arriesgada; no se han hecho intentos en el mundo de obtener un software tal que funcione en teléfonos de gama media, puesto que se considera que J2ME no es capaz de proveer las funcionalidades matemáticas necesarias para este tipo de procesos. Sin embargo, tal como se aprecia en ClubJava [ClubJava 2006], los procesadores de los actuales teléfonos de gama media tienen tanto poder de procesamiento como un PC de hace cinco o siete años. Tal poder, si bien no es abundante, es suficiente para efectuar cálculos livianos. Debemos considerar, además, el problema del pequeño tamaño del heap de memoria de los teléfonos; en el caso del Motorola SLVR L6, por ejemplo, es tan sólo de 800 Kb.

Tal escasez de poder de cálculo hace más agudo el problema de resolver los algoritmos anteriormente revisados. El cálculo de convoluciones incluye, por ejemplo, una sumatoria $O(nm)$, donde n es el tamaño de la ventana de convolución y m el tamaño en píxeles de la imagen.

Vistas las limitaciones de cálculo de los dispositivos móviles, el desarrollo de un software de procesamiento fotográfico para teléfonos celulares parece ser, al menos, una tarea difícil de llevar a cabo. Aún más, las restricciones impuestas por la API de Java MIDP2 sugieren buscar métodos adicionales para permitir la implementación de las funcionalidades necesarias. El desafío, entonces, es poder obtener resultados de manera ágil, de modo de hacer sentir cómodo al usuario del software propuesto, restringiéndose a la cantidad de memoria y CPU disponible en el teléfono.

4.- Trabajo Realizado y Metodología

El grueso del trabajo en este proyecto se concentra en éste capítulo. Se exponen aquí tanto el desarrollo de las pruebas preliminares y sus resultados como el desarrollo del prototipo funcional de la aplicación.

4.1.- Análisis Previo al Desarrollo del Software

Las particulares características técnicas de los teléfonos y las API que se usarán en el desarrollo de este proyecto obligan a efectuar un análisis sobre los desafíos computacionales que es necesario enfrentar y métodos para llevar a cabo estas tareas con éxito.

4.1.1.- Consideraciones Matemáticas

El uso de funciones trigonométricas que no se encuentran implementadas en la API exige, tradicionalmente, el cálculo vía series o el uso de matemáticas discretas. Dado que el cálculo de una serie hasta un n -ésimo exponente es una operación $O(n \cdot \log(n))$ usando programación dinámica, se consideró que esta opción no era adecuada para el uso en un teléfono móvil. Por otra parte, la matemática discreta permite el calcular de manera muy barata las funciones deseadas, a costa de la precisión del cálculo. En particular, la implementación de la matemática discreta en esta memoria permite calcular funciones trigonométricas con costo $O(1)$, limitando la precisión angular a 5° . Esto se hizo posible precargando los valores de $\cos(x)$ y $\arctan(x)$ para cada ángulo entre 0° y 360° , con avances cada 5° , y guardando los valores en una tabla de hashing, que pertenece a una entidad común a toda la aplicación. Si un ángulo cae dentro del rango de 5° de precisión de la tabla de hashing, es interpolado linealmente hasta obtener 1° de precisión.

La aplicación de la matemática discreta permite, por ejemplo, que el costo de rotar una imagen sea $O(n)$ con respecto a la cantidad de píxeles de la imagen.

Aunque originalmente se pensaba implementar rotaciones en ángulos predefinidos (30° , 45° , 60° , 90° , más sus reflexiones en el eje x e y) de manera de disminuir el costo computacional, la implementación efectuada de la matemática discreta permitió el cálculo de rotaciones en ángulo libre con el mismo costo que las rotaciones en ángulos predefinidos.

4.1.2.- Consideraciones Técnicas

Los requerimientos funcionales del software a desarrollar en esta memoria, desde un principio, limitaron fuertemente el espectro de dispositivos móviles para los cuales se podría portar aunque, por supuesto, a largo plazo esto no es un problema grave. Una primera selección limitó la gama de teléfonos objetivo sólo a aquellos camera-phones habilitados para usar la API Java MIDP2. Asimismo, el requerimiento no funcional de costo máximo del teléfono eliminó una gran cantidad de teléfonos candidatos que, si bien soportaban técnicamente al software, no pertenecen a la categoría "gama media". El resultado de esta selección se puede apreciar en la Tabla 1 (página 6).

Es necesario acotar que esta primera selección está orientada completamente por criterios de mercado, puesto que se pudo haber orientado el desarrollo del software a dispositivos de gama alta, desapareciendo en gran parte las limitaciones técnicas de los teléfonos como causal de eliminación. Sin embargo, el objetivo central de esta memoria es que el software que aquí se desarrolle funcione sobre dispositivos de gama media.

El requerir el uso del formato JPEG conllevó la necesidad de usar la API JSR-135 MMAPI. El impacto sobre la cantidad de teléfonos objetivo no fue importante, al encontrarse esta API prácticamente estandarizada en los dispositivos con cámara integrada.

Existen tres maneras de acceder a archivos dentro de un dispositivo móvil usando Java:

1. Utilizando recursos gráficos presentes en el jar.
2. Utilizando recursos gráficos existentes en un servidor vía wap.
3. Utilizando el sistema de archivos del teléfono.

Considerando que el software a crear en esta memoria debe acceder a las imágenes tomadas con la cámara integrada, la única manera posible para lograrlo es accediendo al sistema de archivos del teléfono. Sin embargo, esto no es posible de manera nativa, debiéndose utilizar la API JSR-75 FileConnection, o la API de Siemens. Se optó por el uso de JSR-75, postergando para una posible evolución del software el uso de la API de Siemens, dado que JSR-75 es utilizada por una mayor cantidad de fabricantes de teléfonos.

Lamentablemente, JSR-75 es soportada por un número pequeño de dispositivos, puesto que originalmente fue ideada como un paquete opcional para proveer funcionalidades de PDA a smartphones. El uso de esta API obligó a limitar drásticamente la cantidad de teléfonos disponibles para porting, pasando de tener aproximadamente 10 modelos de teléfonos disponibles a solamente uno, el Motorola L6 SLVR. Pensando en futuras evoluciones de este software, se sugirió la necesidad de considerar una cantidad de teléfonos de gama alta con precio inferior a \$100.000, para futuras pruebas. Estos teléfonos se pueden apreciar en la Tabla 3 (página 6).

4.2.- Benchmarks

Paralelamente a la investigación bibliográfica y la redacción de este informe, se han definido los distintos algoritmos implementados en esta memoria, y que se exponen en extenso en el capítulo anterior. Luego, se ha definido una serie de benchmarks que se desea realizar sobre los teléfonos objetivos y que tienen por objeto predecir el comportamiento de estas técnicas una vez que estén funcionando en los dispositivos móviles. Para esto, se cargaron desde el sistema de archivos del teléfono imágenes de 640X480 píxeles, sobre las cuales se aplicaron, selectivamente, las siguientes operaciones:

- Obtención de histograma
- Estrechamiento de contraste
- Aplicación de una función de convolución utilizando una ventana de convolución de 3X3
- Eliminación de ruido
- Rotación en ángulo libre
- Reconstrucción de la imagen

Con el fin de desarrollar los benchmarks, se ha levantado un sistema compuesto del compilador Java 1.4.2.04, JRE 5.0 y J2ME Wireless Toolkit 2.5 [Sun 2005]. El desarrollo se ha efectuado sobre la IDE Eclipse 3.1 usando el plugin EclipseME 1.5.3 [EclipseMe 2006].

Sobre este sistema se ha creado el esqueleto de la aplicación Benchmarks, el cual está compuesto de las siguientes componentes, tal como se encuentran esquematizadas en el diagrama de clases 1:

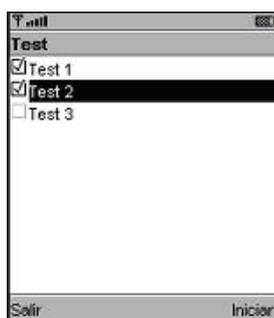


Imagen 12: Aplicación de benchmarks

- Clase Benchmark (main, MIDlet): Pantalla principal, permite elegir entre las distintas pruebas a ejecutar mediante un control Select múltiple. Ejecuta las pruebas seleccionadas en forma secuencial.
- Clase TestManager: Recibe los valores entregados por el control Select de Benchmark y lanza los tests seleccionados, secuencialmente, sobre nuevos threads.
- Clase Test: Clase abstracta que debe ser implementada por cada uno de los tests.

- Clases TestX (MIDLet): Implementaciones de la clase Test; cada una de estas clases contiene las instrucciones para efectuar sólo un benchmark.
- Clase mathProcessor: Implementa un objeto capaz de procesar eficientemente cálculos matemáticos fuera de las posibilidades de MIDP2.
- Clase ImageFileManager: Implementa un objeto encargado tanto de recibir imágenes desde otras clases y grabarlas al sistema de archivos como de abrir imágenes desde el sistema de archivos y entregarlas a otras clases.
- Clase ImageProcessor: Aplica las transformaciones a las imágenes.

Para transferir las aplicaciones a los teléfonos, se han utilizado un adaptador Bluetooth para PC.

4.2.1.- Resultados de benchmarks

Sobre el teléfono Motorola L6 SLVR se midió el tiempo que demoró el programa en efectuar las operaciones solicitadas, tomando el promedio sobre diez mediciones. Los resultados de estas pruebas se exponen en la Tabla 4:

Tabla 4: Resultado de Benchmarks

Prueba	Tiempo Promedio [s]
Rotación en Angulo Libre	21,2
Eliminación de Ruido	43,2
Obtención de Histograma	18,5
Estrechamiento de Contraste	20,8
Función de Convolución	43,8
Grabación de Imagen	3,2

Es importante acotar el que, considerando las estrictas restricciones que imponen los teléfonos sobre los tamaños de los jar, los cuales por razones comerciales y técnicas deben ser lo más pequeños posibles, el tamaño final de la aplicación de benchmarks, no superior a los 25 Kb, sugiere un acierto en la arquitectura de la aplicación.

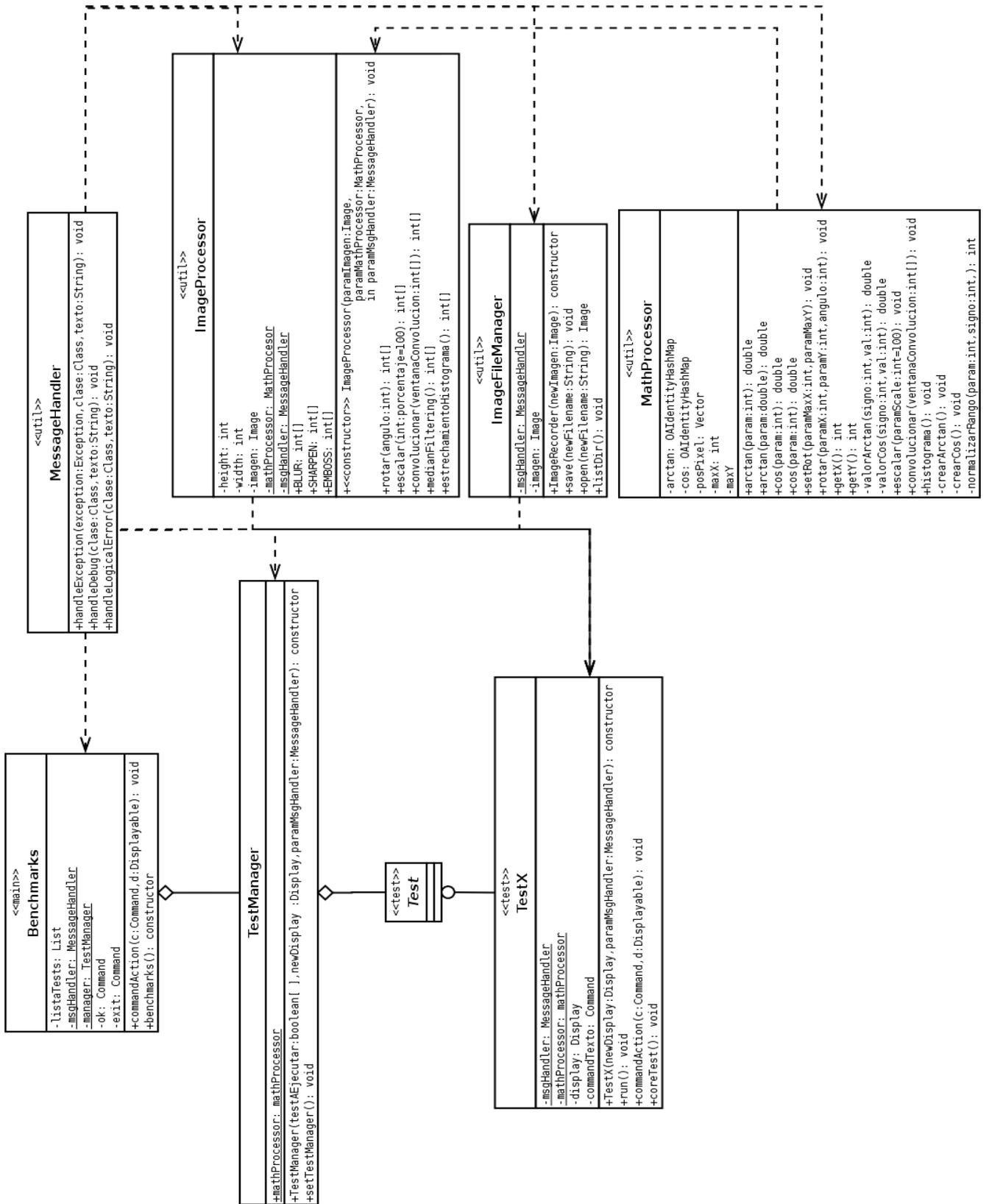


Diagrama de Clases 1

4.3.- Desarrollo

Una vez finalizados los benchmarks, el siguiente paso es la definición y optimización de las técnicas a incorporar al prototipo funcional y la codificación de éste.

4.3.1.- Creación de un prototipo funcional

Se desarrolló un prototipo funcional de la aplicación usando el lenguaje J2ME MIDP2 [Sun 2005]. Se usó el mismo sistema computacional usado para la construcción de la aplicación de benchmarks.

Para facilitar el porting se ha utilizado la herramienta opensource J2MEPolish 2.0 beta, la cual se compone de una serie de archivos de configuración de tareas Ant que permiten crear gran cantidad de portings automáticamente, optimizando el código intermedio y ofuscando los jar finales. Asimismo, ha sido necesario usar la API JSR-135 MMAPI para utilizar las funciones relativas a la cámara fotográfica y a los códecs JPEG, y a la API JSR-75 Fileconnection - PIM API para poder acceder al uso del sistema de archivos nativo de los dispositivos móviles.

El prototipo funcional reutiliza la estructura creada para realizar los benchmarks, puesto que ésta fue creada modularmente y de modo tal que fuese sencillo modificar o agregar nuevas funcionalidades. Sin embargo, ha sido necesario relajar un poco la orientación a objetos en la arquitectura de la aplicación, tal como es recomendado por muchos autores de artículos sobre desarrollos J2ME, de modo de economizar recursos computacionales. Esto se refleja claramente en el Diagrama de Clases 2, el cual expone tanto la estructura de clases como el flujo de la información del prototipo funcional.

Este prototipo incorpora nuevas funcionalidades, no presentes en la aplicación de benchmarks. Entre estas, se encuentran:

- Navegador del sistema de archivos del teléfono.
- Lectura de imágenes reales y creación de thumbnails.
- Grabación de imágenes.

Para poder grabar las imágenes al sistema de archivos del teléfono fue necesario escribir un codificador PNG. Se seleccionó este formato debido a que su codificación es mucho más simple que la del formato JPEG, y porque el formato PNG es soportado nativamente por J2ME.

Los iconos creados se muestran en la Tabla 5, y las imágenes 13 a 19 corresponden a las interfaces. Es necesario notar que todas las interfaces son muy parecidas entre sí, de manera de evitar al usuario el tener que aprender a usar más controles de los estrictamente necesarios. La jerarquía de interfaces se expone en el Diagrama 1.

Icono	Significado	Ámbito de existencia
	Icono de la aplicación	Menú del teléfono
	Archivo que no es una imagen	Navegador del filesystem
	Archivo de Imagen	Navegador del filesystem
	Carpeta	Navegador del filesystem
	Subir a carpeta superior	Navegador del filesystem
	Menú Rotar / Regulación manual de rotación	Operaciones / Menú Rotar
	Rotar en 90°	Menú Rotar
	Rotar en -90°	Menú Rotar
	Rotar en 180°	Menú Rotar
	Menú Tamaño / Aumento de tamaño 200%	Operaciones / Menú Tamaño
	Reducción de tamaño 50%	Menú Tamaño
	Redimensionamiento manual	Menú Tamaño
	Menú brillo / Regulación manual	Operaciones / Menú Brillo
	Reducción de brillo 50%	Menú Brillo
	Aumento de brillo 200%	Menú Brillo
	Menú Contraste / Aumento de contraste 200%	Operaciones / Menú Contraste
	Reducción de contraste 50%	Menú Contraste
	Regulación manual	Menú Contraste
	Suavizar	Operaciones
	Endurecer	Operaciones
	Blanco y negro	Operaciones
	Negativo	Operaciones
	Repujado	Operaciones
	Deshacer operación anterior	Operaciones
	Grabar imagen a filesystem	Operaciones
	Información sobre la imagen	Operaciones

Tabla 5: Iconos

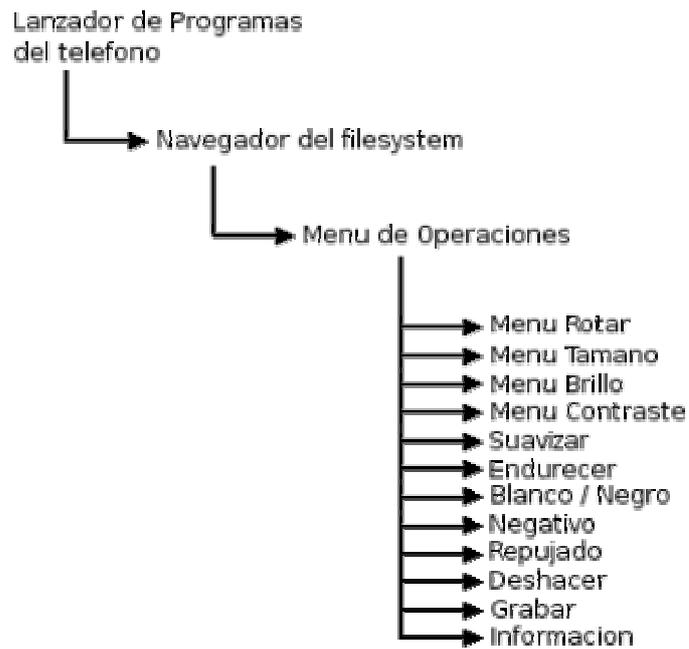


Diagrama 1: Estructura de menús de la aplicación

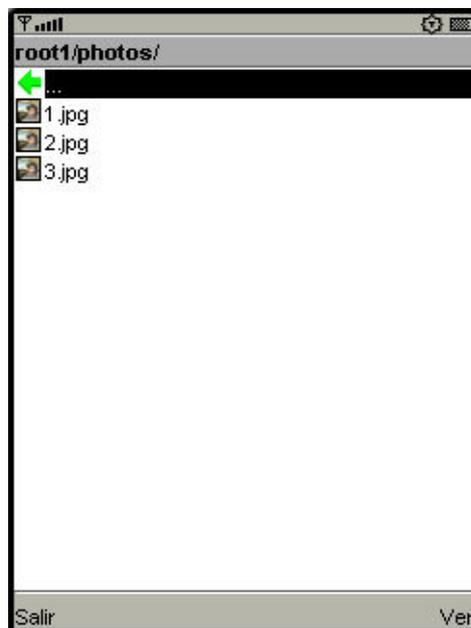


Imagen 13: Screenshot del navegador del filesystem

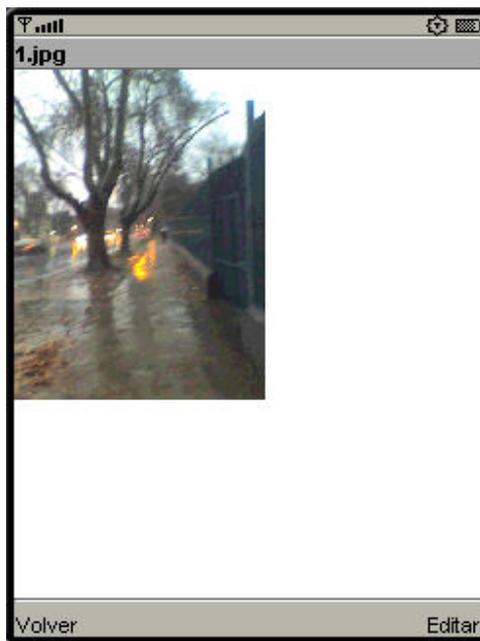


Imagen 14: Interfaz de edición de imagen



Imagen 15: Interfaz del menú de operaciones



Imagen 16: Interfaz del menú de Tamaño

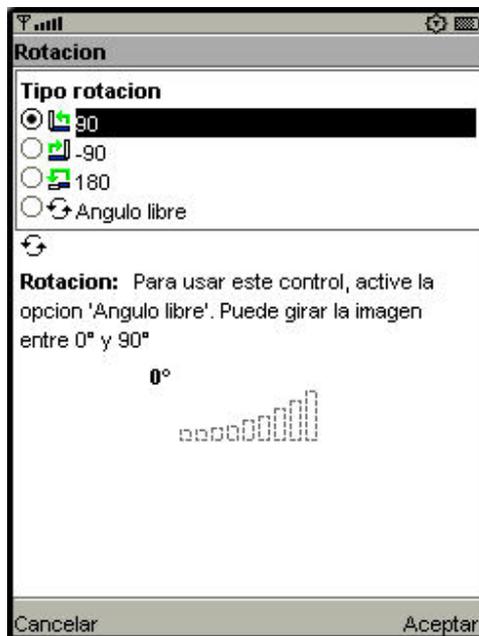


Imagen 17: Interfaz del menú de Rotación



Imagen 18: Interfaz del menú de Brillo



Imagen 19: Interfaz del menú de Contraste

4.3.2.- Técnicas implementadas y optimizaciones

Usando los algoritmos mencionados en la revisión bibliográfica, se implementaron en esta memoria variaciones de las siguientes técnicas:

- Corrección Gamma.
- Ecuación de histograma.
- Suavizado (blur).
- Endurecimiento (sharpen).
- Repujado (emboss).
- Escalamiento.
- Rotación.

Se implementaron, adicionalmente, las siguientes técnicas:

- Transformación a Escala de Grises.
- Negativo.

Fue necesario, a fin de optimizar la velocidad de la aplicación, sacrificar la precisión de las técnicas originales aplicando las siguientes aproximaciones:

- **Corrección Gamma:** Se evitó el tener que elevar a la potencia $1/\gamma$ aplicando un algoritmo de manejo de brillo que consiste en multiplicar el valor de los canales RGB por el porcentaje de brillo deseado. Este algoritmo fue ideado por el autor; sin embargo, posteriormente se comprobó que esta es una manera relativamente común de efectuar esta operación.
- **Contraste:** Un algoritmo que se comporta de manera similar a la ecuación de histograma consiste en tomar el valor de cada canal RGB, convertirlo a valor porcentual donde 0% es el valor 0 y 100% es el valor 255, se resta un 50%, se multiplica por el modificador de contraste deseado, se suma 50% y se vuelve a la escala de valores entre 0 y 255 [Dusina 2006].

Este algoritmo no garantiza el estiramiento del histograma y la calidad de las imágenes, aunque es inferior a las obtenidas mediante la ecuación de histograma, es aceptablemente buena.

Las modificaciones a los algoritmos de corrección gamma y de contraste evitaron por completo el tener que usar tablas de consulta. Asimismo, dado que los nuevos algoritmos trabajan exclusivamente sobre el espacio de colores RGB, se evitó el tener que convertir los colores al espacio HSI. Además se evita una lectura en la imagen, puesto que usando tablas de consulta se debe leer la imagen una vez para crear la tabla de consulta y otra vez para realizar la transformación desde la imagen original a la imagen final, pasando por la tabla de consulta; en cambio, usando el algoritmo modificado, sólo es necesario leer la imagen original una sola vez.

La transformación a escala de grises se realizó usando transformaciones de colores nativas de J2ME.

La imagen en negativo fue obtenida reemplazando cada componente RGB por 255 – el valor original del canal RGB.

Se determinó, mediante el estudio de distintas implementaciones, que la reducción de ruido mediante median filtering es demasiado cara, en términos de cómputo, para un teléfono. Se prefirió, en su lugar, implementar solamente el filtro de suavizado, que puede actuar como filtro de eliminación de ruido, al costo de perder precisión en los detalles.

Los filtros de convolución fueron desarrollados usando templates de 3X3. Se convolucionó sobre los tres canales RGB, sumando los valores del template sobre los cuales se ha evolucionado para cada píxel. Este valor se usó como divisor en los valores RGB obtenidos tras la convolución, de modo de compensar los valores RGB obtenidos al reconstruir el nuevo color [Ackers 2006].

Los templates que se usaron para implementar los distintos filtros de convolución son:

- **Suavizado:**
$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$
- **Endurecimiento:**
$$\begin{bmatrix} 1 & 1 & -1 \\ 1 & 4 & -1 \\ 1 & -1 & -1 \end{bmatrix}$$
- **Repujado:**
$$\begin{bmatrix} -2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -2 \end{bmatrix}$$

Si bien las matrices de suavizado y endurecimiento son bien conocidas en la literatura [Ackers 2006], es difícil encontrar buenas matrices de repujado. La anterior fue escogida por el autor tras experimentar con algunas matrices propuestas hasta descubrir una matriz cuyos resultados fuesen satisfactorios.

La creación de thumbnails, si bien no es obvia al usar la aplicación, es fundamental al crear las imágenes a escala que se muestran al usuario.

4.3.3.- Testing

Para testear la aplicación se usó un teléfono Motorola L6 SLVR. El primer problema que apareció en la etapa de testing es que, para poder usar JSR-75 en los teléfonos reales, los jar deben estar firmados por alguna entidad reconocida por Motorola. Para poder obtener una firma, existen dos caminos a seguir:

1. Establecer una relación de negocios con Motorola.

2. Enviar la aplicación a testing a alguna empresa de QA asociada a [JavaVerified 2006] The Java Verified Program.

La primera opción es la mejor si se pretende seguir trabajando en desarrollos J2ME sobre teléfonos Motorola; el autor se encuentra actualmente en conversaciones con Motorola al respecto. Sin embargo, este proceso es largo y complejo, y por definición está limitado a empresas desarrolladoras. La segunda opción es impracticable para un alumno de ingeniería, puesto que los testings más baratos tienen un costo que ronda los US\$750 ([AndinaTech 2006],[JavaVerified 2006]).

Para poder probar la aplicación sobre un dispositivo real, se ha creado una versión del prototipo funcional que no usa JSR-75, sino que tiene algunas imágenes dentro del jar. Si bien esta versión no es distribuible tanto porque su peso es excesivo (146 Kb) como porque no permite modificar archivos del sistema de archivo, lo cual es central en el planteamiento de esta memoria, esta versión procesa las imágenes de manera exactamente igual que la versión original del software. Con esta versión se probaron todas las funcionalidades del software (exceptuando, por supuesto, aquellas relativas al sistema de archivos, como grabar imágenes), obteniendo los siguientes resultados como promedio de 10 corridas sucesivas:

Operación \Tamaño de la imagen	Tiempo promedio [s]	
	320X240	640x480
Rotación 16°	55	202
Redimensionamiento 50%	7	26
brillo	16	61
contraste	21	82
suavizar	72	268
endurecer	75	297
blanco/negro	14	52
Negativo	13	55
Repujado	76	300
Deshacer	4	13

Tabla 6: Tiempo promedio de ejecución de operaciones en ambiente de testing

Es necesario acotar que el redimensionamiento depende completamente del área de la imagen objetivo. Un redimensionamiento al 200% demora aproximadamente cuatro veces más que una operación lineal sobre la imagen original, tal como brillo o contraste.

Se muestran, en las imágenes 20 a 23, fotografías que muestran al programa de testing en distintas fases de su trabajo:

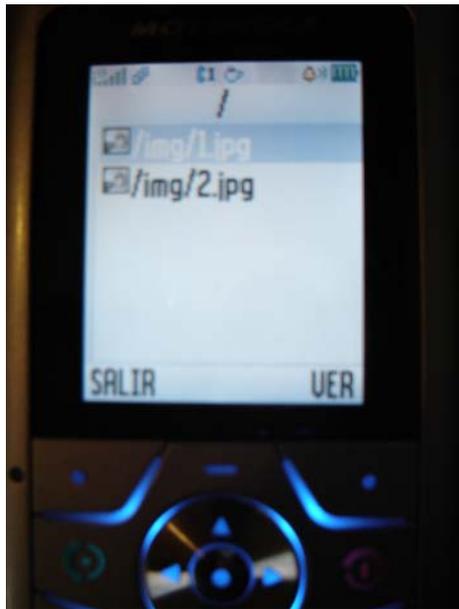


Imagen 20: Navegador de filesystem modificado para leer imágenes desde el jar

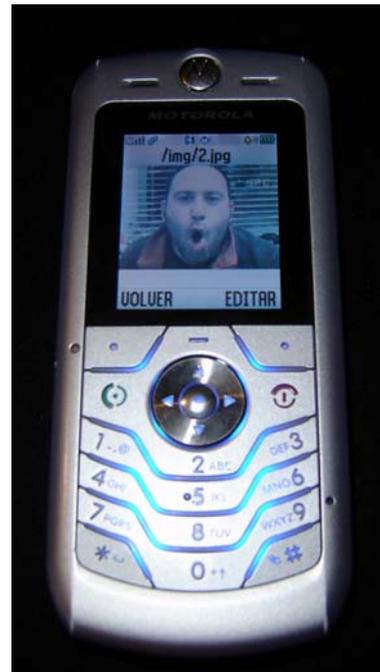


Imagen 21: Display de imágenes



Imagen 22: Toma de datos



Imagen 23: Menú de Rotación

A continuación, se exponen diferentes imágenes procesadas con el prototipo funcional, obtenidas usando el emulador de J2ME. Estas imágenes se comparan con sus respectivos originales y con imágenes procesadas de manera similar pero usando el software Corel Paint Shop Pro IX.



Imagen 24: Izquierda: Imagen original. Centro: 50% brillo. Derecha: Paint Shop Pro

Inmediatamente se observa una cierta discretización de la paleta de colores disponible al cargar la imagen. Este fenómeno se produce por las aproximaciones en los sampleos de colores que J2ME debe hacer; es parte del trabajo a futuro el eliminar tales problemas.

Se observa que el manejo del brillo se comporta de manera muy similar al del software profesional Paint Shop Pro IX.



Imagen 25: Izquierda: Imagen original. Centro: 50% contraste Derecha: Paint Shop Pro



Imagen 26: Izquierda: Imagen original. Centro: 200% contraste. Derecha: Paint Shop Pro

Observando las imágenes 25 y 26 se puede deducir que el algoritmo modificado de manejo de contraste se comporta de manera excelente, funcionando de manera muy similar al algoritmo comercial utilizado en Paint Shop Pro.



Imagen 27: Izquierda: Imagen original. Centro: Suavizar. Derecha: Paint Shop Pro (Blur)



Imagen 28: Izquierda: Imagen original. Centro: Endurecer. Derecha: Paint Shop Pro (Sharpen more)

El filtro de suavizado implementado es un poco más suave que el filtro de blur que se encuentra en Paint Shop Pro. Sin embargo, el filtro de endurecimiento es mucho más fuerte que su equivalente de Corel. Para lograr un resultado relativamente equivalente se ha debido utilizar el filtro más poderoso de la gama Sharpness de Paint Shop Pro, Sharpen more.

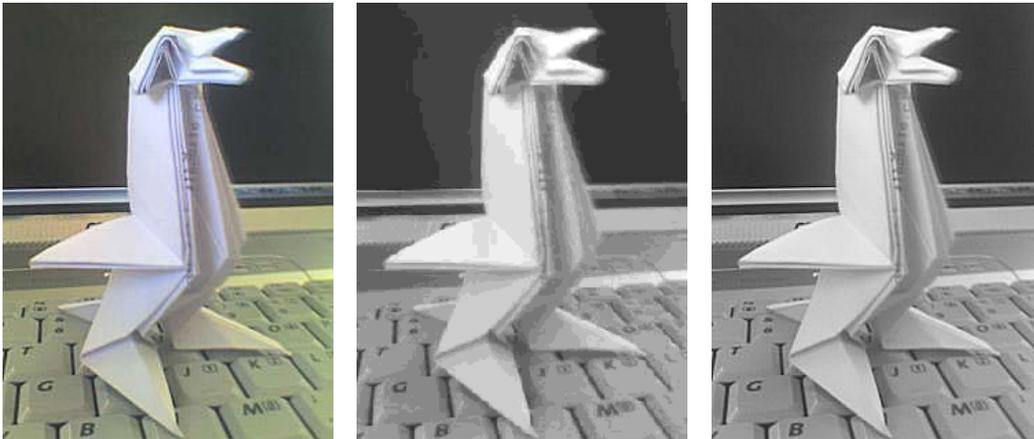


Imagen 29: Izquierda: Imagen original. Centro: Escala de Grises. Derecha: Paint Shop Pro

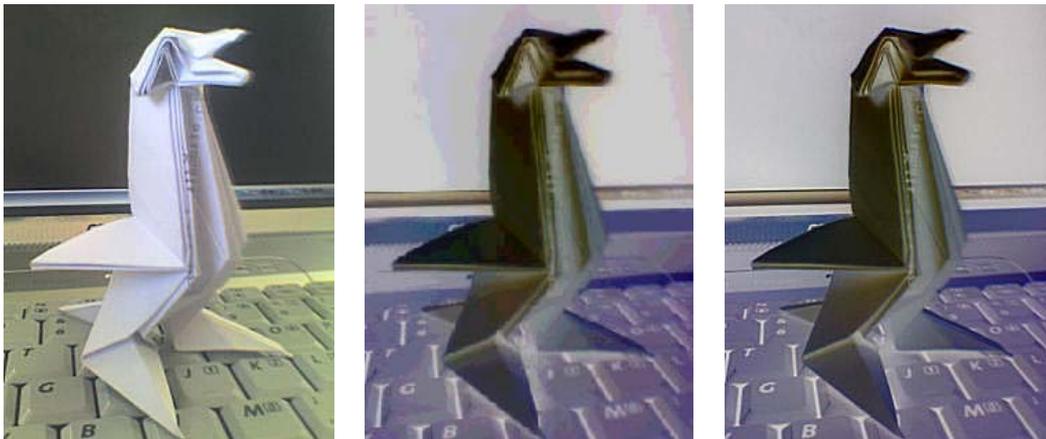


Imagen 30: Izquierda: Imagen original. Centro: Negativo. Derecha: Paint Shop Pro

Los efectos de negativo y suavizado funcionan exactamente igual que en su equivalente de Corel; la única diferencia observable es la discretización de colores antes mencionada.



**Imagen 31: Izquierda: Imagen original. Centro arriba: Repujado. Derecha: Paint Shop Pro (Emboss).
Abajo: Paint Shop Pro (Black Pencil/Medium Pencil)**

El filtro Repujado no tiene un equivalente exacto en Paint Shop Pro, pudiéndose establecer cierta similitud con el filtro Emboss. En esta imagen, se compara tanto con Emboss como con el filtro Black Pencil/Medium Pencil.

4.4.- Inserción en el Mercado

Dado que el único sistema de aprovisionamiento de contenido multimedia por parte de las operadoras a sus clientes es vía OTA (transporte por señales telefónicas aéreas) y considerando que una exitosa inserción y posible campaña de mercadeo externa depende fuertemente del apoyo oficial de las operadoras, se descarta el distribuir este software por venta directa de CD o descarga al PC. Esto nos lleva a considerar dos opciones para la venta y distribución:

1. Construcción de un sitio WAP propio.
2. Asociación con un publicador de contenidos para telefonía celular.

Si bien la primera opción permite libertad total sobre la masa de clientes a la que se puede impactar, permitiendo el acceder a la aplicación a la población de usuarios de camera-phones mundial completa, se hace prácticamente imposible el contar con el apoyo de las operadoras telefónicas y, aún peor, elimina prácticamente la opción de que éstas colaboren dando a conocer el software. Es decir, se accede a una gran cantidad de usuarios, pero no existen maneras asequibles de promocionar la aplicación. Además, el dueño de la aplicación debería gastar recursos contratando hosting, comprando un dominio y asociándose con métodos de pago online, tales como transbank o pay-pal.

La segunda opción tiene como desventaja el que cada publicador cuenta con una cartera de clientes limitada; sin embargo, el propio distribuidor y eventualmente la operadora, si se encuentra interesada, se encargarán de promocionar la aplicación. El creador de la aplicación no necesita proveer de hosting, dominio ni sistema de cobro, puesto que la distribuidora se encarga de tales problemas. El precio de venta de la aplicación, por supuesto, debe subir, puesto que habría más actores involucrados en la venta de la aplicación.

Un beneficio adicional de esta opción es que una distribuidora bien conocida puede fácilmente establecer relaciones de negocios con los fabricantes de terminales como Motorola, con lo cual se hace muy fácil el obtener firmas para los jars. El conseguir firmas como profesional independiente se hace muy caro de otra manera, debiéndose contratar como testers a empresas asociadas a The Java Verified Program [JavaVerified 2006].

Considerando los costos y beneficios, se ha decidido que la mejor opción es la segunda. Para este fin, se ha hecho contacto con la empresa AndinaTech S.A. [AndinaTech 2006], publicadora de juegos y contenido multimedia para dispositivos móviles, contemplando una eventual distribución del software. El modelo de negocios a adoptar sería el revenue share, donde el total del cobro al usuario final se divide en un 40% para la operadora, un 30% para la publicadora y un 30% para el creador. Usando a AndinaTech como publicador, se puede eventualmente acceder a los usuarios de las operadoras Movistar Chile, Oí, Amazonia y Telemig en Brasil, Movistar Argentina, Movistar Uruguay y, eventualmente, Claro Chile y Brasil. Además, se cuenta con los portales Terra.cl, Juegos.Movistar.com y el portal Los Fanáticos, propiedad de AndinaTech.

5.- Resultados y Conclusiones

Las conclusiones de este proyecto apuntan a definir posibles desarrollos futuros del software obtenido, así como investigar métodos para una posible inserción en el mercado de la aplicación final.

5.1.- Software Obtenido

Como resultado de esta memoria, se ha obtenido un software desarrollado en Java J2ME MIDP2 capaz de efectuar procesamiento in-situ a las fotografías tomadas en camera-phones. Si bien las limitantes técnicas y económicas de los teléfonos restringen fuertemente la cantidad de dispositivos para los cuales esta aplicación se ha desarrollado, la tendencia de la industria indica que en corto plazo existirán muchos dispositivos dentro del límite de precio propuesto capaces de usar esta aplicación. Asimismo, existen teléfonos que, aunque cuestan más de \$80.000, tienen un precio no muy superior a esta cifra, lo que los convierte en posibles candidatos para futuros portings. Este es el caso, por ejemplo, del Motorola V555, Nokia 6101 y Sony-Ericsson K510.

Asimismo, es posible portar, usando las API's de extensión propietarias de BenqSiemens, para los teléfonos BenqSiemens de la generación 65/70, tales como los populares BenqSiemens C65, C66 y M65. La inclusión de estos modelos permite a la aplicación estar disponible para el segmento de usuarios más atractivo en términos de mercado [AndinaTech 2006].

Existe, sin embargo, la necesidad de obtener una firma autorizada para el jar, sin la cual el software no podrá hacer uso del sistema de archivos de los teléfonos ni, por lo tanto, acceder a los contenidos de la cámara. El autor se encuentra actualmente en conversaciones con Motorola por este tema, no descartándose el asociarse a una publicadora de software.

5.2.- Desarrollos Futuros

Futuros desarrollos del prototipo funcional apuntarían a convertir el prototipo en una aplicación madura para el mercado. Se ha definido que es necesario tratar puntos tales como:

- Agregar la posibilidad de grabar imágenes en formato JPG.

- Eliminar la discretización de colores presente en el prototipo funcional.
- Agregar cuadros de diálogo que muestren al usuario que la aplicación está trabajando, de modo que éste no crea que el teléfono se ha colgado.
- Mejorar la interfaz de usuario, para hacerla aún más accesible.
- Mejorar el porting. Para esto, se hace necesario usar API propietarias, de modo de hacerla accesible al mercado de usuarios de teléfonos masivos Siemens, Motorola, Nokia y Sony-Ericsson, quienes fabrican la gran mayoría de los camera-phones masivos.
- Revisar los algoritmos, de modo de intentar ahorrar aun más procesamiento.
- Estudiar métodos de procesamiento automático de imágenes, evitando a los usuarios el tener que decidir que proceso es mejor para sus imágenes.

Como futuros trabajos no relacionados a la codificación, se consideran:

- Establecer relaciones comerciales con los fabricantes de terminales.
- Establecer relaciones comerciales con publicadoras de software para dispositivos móviles.

6.- Bibliografía

[About 2006] About.com (2006), Cellullar Phones Mobile Phones Mega Information Center at About.com, <http://cellphones.about.com/>, 16 de Junio de 2006.

[Ackers 2006] Fred Ackers (2006), CodeProyect.com, Convolution of bitmaps, <http://www.codeproject.com/bitmap/ImageConvolution.asp>, 2 de Noviembre de 2006.

[Almalence 2006], Almalence Inc. (2006), PhotoAcute Suite, <http://www.almalence.com>, <http://www.photoacute.com>, 20 de Junio de 2006.

[AndinaTech 2006] AndinaTech (2006). Comunicación personal con Ana Paola Teixeira, Gerente General, anapaola@andinatech.com.

[BenqSiemens 2006] Benq-Siemens (2006), Benq-Siemens Portal, <http://www.benq-siemens.com>, 26 de Junio de 2006.

[BenqSiemens Developer 2006], Benq-Siemens 2006, Sitio oficial para desarrolladores, <https://market.benqmobile.com>, 18 de Noviembre de 2006.

[Bitween 2006] Bitween (2006), Photo Editor, <http://www.bitween.com/sito/catalog.php?genre=2&but=1>, 18 de Abril de 2006.

[CPR 2006] Camera Phone Reviews (2006), Camera Phone Reviews, <http://www.livingroom.org.au/camera-phone/>, 16 de Junio de 2006.

[Claro 2006] Claro Chile (2006), Portal de Claro Chile, <http://www.clarochile.cl>, 27 de Septiembre de 2006.

[ClubJava 2006] Club Java (2006), MIDP Java Phone Benchmarks, http://www.club-java.com/TastePhone/J2ME/MIDP_Benchmark.jsp, 18 de Abril de 2006.

[Cnet 2006] Cnet Reviews (2006), Cell phones: Cell phone reviews, Mobile phone reviews, Wireless phones, http://reviews.cnet.com/Cell_phones/2001-3504_7-0.html, 16 de Junio de 2006.

[Digia 2006] Digia (2006), ImagePlus, <http://www.sysopendigia.com>, 24 de junio de 2006.

[Dusina 2006] Mitch Dusina (2006). developerfusion.co.uk, Image manipulation – brightness and contrast, <http://www.developerfusion.co.uk/show/5441/3/>, 3 de Noviembre de 2006.

[EclipseMe 2006] EclipseMe (2006), EclipseMe Eclipse Plugin, <http://eclipseme.org/>, 4 de Julio de 2006.

[ENTEL 2006] ENTEL PCS (2006), Listado de Equipos, <http://www.ENTELpcs.cl/>, 16 de Junio de 2006.

[Escalera 2001] Arturo de la Escalera Hueso, Visión por Computador, Fundamentos y Métodos, Prentice Hall (2001).

[Fisher 2003] Robert Fisher, Simon Perkins, Ashley Walker y Erik Wolfart (2003) Hypermedia Image Processing Reference: Spatial Filters, <http://homepages.inf.ed.ac.uk/rbf/HIPR2/median.htm>, 5 de Julio de 2006.

[Google 2006] Google (2006), Picasa, <http://picasa.google.com>, 4 de julio de 2006.

[GSMarena 2006] GSM Arena (2006), GSM reviews, opinions, votes, manuals, ringtones and more..., <http://www.gsmarena.com/>, 16 de Junio de 2006.

[JavaVerified 2006] The Java Verified Program (2006), The Java Verified Program, <http://www.javaverified.com>, 6 de Noviembre de 2006.

[Mai 2006] Mai, Luong Chi, Introduction to Computer Vision and Image Processing (Fecha desconocida), <http://www.netnam.vn/unescocourse/computervision/computer.htm>, 14 de Junio de 2006.

[MobileReview 2006] Mobile Review (2006), Mobile Review, <http://www.mobile-review.com/index-en.shtml>, 16 de Junio de 2006.

[Motodev 2006] MOTODEV (2006), Sitio de Motorola para desarrolladores, <http://developer.motorola.com>, 26 de Agosto de 2006.

[Movistar 2006] Movistar (2006), Listado de Equipos, <http://tienda.movistar.cl/default.asp>, 16 de Junio de 2006.

[MyMob 2006] MyMob Mobile Portal (2006), Mobile Phone Specification, <http://www.mymob.com/mobiles/>, 16 de Junio de 2006.

[Nokia 2006] Nokia (2006) Nokia Photo Editor, <http://europe.nokia.com/nokia/0,,64483,00.html>, 18 de Abril de 2006.

[Nokia Forum 2006] Forum Nokia (2006), Sitio oficial de Nokia para desarrolladores, <http://www.forum.nokia.com>, 13 de Noviembre de 2006.

[Patin 2003] Patin, Frédéric (2003) An Introduction to Digital Image Processing, <http://www.gamedev.net/reference/programming/features/imageproc/page2.asp>, 18 de Abril de 2006.

[PhoneArena 2006] phoneArena.com (2006), Mobile Phone Specifications, Reviews, News And Forum, <http://www.phonearena.com/htmls/home.php>, 16 de Junio de 2006.

[PhoneScoop 2006] Phone Scoop (2006), cell phone news, info, reviews, community, and deals, <http://www.phonescoop.com/>, 2006.

[Phoneyworld 2006] Phoneyworld.com (2006), Mobile Phones, <http://www.phoneyworld.com/handsets/index.aspx>, 16 de Junio de 2006.

[PNG 1996] libpng.org (1996) Portable Network Graphics Specifications 1.2, <http://www.libpng.org/pub/png/spec/1.2/PNG-Contents.html>, 8 de Diciembre de 2006.

[Polish 2006] J2ME Polish (2006) J2ME Polish, <http://www.j2mepolish.org/>, 18 de Abril de 2006.

[Richter 1999] Richter, Reiner (1999) Convolution Filters Research, <http://members.optusnet.com.au/wardead/filters/>, 18 de Abril de 2006.

[Softex 2006] Softex Digital (2006) Photo Editor, <http://www.softexdigital.com>, 18 de Abril de 2006.

[Softpedia 2006] Softpedia (2006) Gsm & Mobile News, <http://mobile.softpedia.com/>, 16 de Junio de 2006.

[Sony Ericsson 2006] Sony Ericsson (2006), Sitio oficial para desarrolladores de Sony Ericsson, <http://www.developer.sonyericsson/>, 13 de Noviembre de 2006.

[SUBTEL 2005] Subsecretaría de Telecomunicaciones (2005) Estadísticas del Desempeño del Sector de las Telecomunicaciones en Chile: Junio 2004 – Junio 2005, http://www.subtel.cl/pls/portal30/docs/FOLDER/WSUBTEL_CONTENTIDOS_SITIO/SUBTEL/ESTDEMERCADO/INFESTAD/INFESTAD2/INFO_ESTADISTICO_10_JUN.PDF, 18 de Abril de 2006.

[Sun 2005] Sun Microsystems (2005) Java Platform, Micro Edition (ME), <http://java.sun.com/javame/>, 18 de Abril de 2006.

[WildPalm 2006] Wild Palm (2006) Camera FX, <http://www.wildpalm.co.uk>, 20 de Junio de 2006.

[Yaoo 2006] Yaoo.net (2006), FlexiCam, <http://www.yaoo.net>, 20 de Junio de 2006.

[Zenzis 2006] Zenzis (2006) PhotoRite SP, http://www.zensis.com/photorite_sp/main.aspx?lang=en&, 18 de Abril de 2006.