

**Universidad de las Ciencias Informáticas**

**Facultad 10**



**Título: Implementación de algoritmos para la detección y localización de texto en imágenes.**

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

**Autores:**

Yanet Cesaire Velázquez  
Ana Julia Fonseca Avila

**Tutora:** Ing. Suset Fernández Rojas  
**Co-Tutor:** Gilberto Lissabet Hernández

**Ciudad de la Habana, Junio 2009**



*Aprendí que no se puede dar marcha atrás,  
que la esencia de la vida es ir hacia adelante.*

*Agatha Christie*

*El camino hasta aquí ha sido largo, lleno de obstáculos y experiencias. Agradezco hoy a mi compañera Yanet por su constante dedicación y abnegación y a nuestra tutora por su apoyo. A mi hermana por ser mi guía y ejemplo a seguir, a mi pareja por su constante comprensión y cariño, a mis amigos por levantarme cada vez que caí. A todos ustedes que, de una u otra forma, convirtieron mi sueño en realidad.*

Ana Julia

*Antes que nada quiero dar gracias a la Virgen de la Caridad del Cobre, por darme fuerza y voluntad para seguir adelante, ya que solo triunfa en el mundo quien se levanta y busca las circunstancias y las crea si no las encuentra.*

*Agradecer a mi tutora por apoyarnos siempre y por su ayuda incondicional. Agradecer a mi compañera de tesis por ser tan buena conmigo y por aceptar hacer la tesis conmigo. Agradecer mientras viva a mi mamá CACHI, la mujer que siempre ha estado a mi lado, dándome apoyo y seguridad. Creo que sin ella nunca hubiera logrado nada. Su dedicación para conmigo hoy se convirtió en nuestro triunfo, TE QUIERO.*

*Siempre le daré gracias a Dios por haberme dado mi más grande tesoro, mi papito, por sus consejos, su dedicación, su paciencia y su amor por mí. Hoy todo valió la pena y creo que parte de mi victoria se la debo a él, que siempre me guió orientó en lo que soy hoy. GRACIAS PAPITO, TE QUIERO MUCHO. Agradecer a mi hermanita por quererme tanto y por brindarme tanto apoyo. TE QUIERO MUCHO MI HERMANA. Agradecer a mi hermanito "BATIDITO DE PLOMO" por demostrar que me quiere a su manera a pesar de tantas maldades que me ha hecho. TE QUIERO MUCHO. Agradecer a mi sobrino "EL CHICHI" por recordarse siempre de su tía querida en todo momento. TE QUIERO MUCHO NENE. Agradecer a toda mi familia que de una forma o de otra me han apoyado mucho. Agradecer a mi gran amiga Mayelín por preocuparse siempre por mí aunque en estos instantes no esté a mi lado.*

*Agradecer a mí amiga Yordailis (el poni) por apoyarme en los momentos difíciles y estar ahí para todo. Agradecer a mi novio el Flaco por quererme tanto y por darme tantos consejos cuando más lo necesitaba. TE QUIERO MUCHO MI FLACO. Agradecer a todos mis compañeros de estudio a Yiri, Chino, Idel, Carlos, Alberto, Eduardo, Ángel, Diana, Dalvis, Karina, Ilearsi y Arlenys.*

Yanet

*A mis padres por su amor y apoyo incondicional.*  
*Ana*

*A mis padres y hermanos por toda la confianza, cariño y respeto  
depositado en mí.*  
*Yanet*

Declaramos ser autores del presente trabajo de diploma y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo, con carácter exclusivo.

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de junio del año 2010

Yanet Cesaire Velázquez

---

Ana Julia Fonseca Avila

---

Suset Fernández Rojas

---

Gilberto Lissabet Hernández

---

Las imágenes han cambiado la visión de las masas. Hoy en día ellas desempeñan un rol importante en la prensa ya que no solo cumplen una función estética en la noticia, sino que constituyen un recurso informativo que contextualiza, amplía y apoya la información textual. En la actualidad la gestión de imágenes constituye un reto en los medios de prensa cubanos. El Centro de Información para la Prensa (CIPRE) cuenta con la ayuda de la Universidad de las Ciencias Informáticas (UCI) para implementar un sistema que permita trabajar con las mismas. El presente trabajo: “Implementación de algoritmos para la detección y localización de texto en imágenes” responde a una de las necesidades existentes en los medios de prensa cubanos. En él se lleva a cabo una profunda investigación de los métodos de detección y localización de texto en imágenes existentes. Es realizada la implementación de dos de estos algoritmos y se ofrece una caracterización del método, lenguajes y herramientas empleadas para el desarrollo de los mismos.

### **PALABRAS CLAVES**

Imagen, detección de texto, detección de bordes, localización

INTRODUCCIÓN .....	1
CAPÍTULO 1:FUNDAMENTACIÓN TEÓRICA .....	5
Introducción.....	5
1.1 Antecedentes .....	5
1.1.1 Centro de Información para la Prensa.....	5
1.1.2 Ecúmene Pyxel .....	6
1.2 Conceptos .....	7
1.2.1 De imagen a imagen digital .....	7
1.2.2 Imagen digital .....	8
1.2.3 Imagen digital a colores .....	8
1.2.4 Imagen digital en escala de grises .....	9
1.2.5 Detección y localización de texto en imágenes .....	9
1.3 Métodos de detección de textos en imágenes .....	10
1.3.1 Métodos basados en la detección de bordes.....	10
1.3.1.1 Detector de Canny .....	11
1.3.1.2 Operador de Sobel y operador de Prewitt.....	12
1.3.1.3 Operador Roberts .....	13
1.3.2 Métodos basados en la detección de textura.....	14
1.3.2.1 Técnica de detección de textura basada en la transformada wavelet.....	15
1.3.2.2 Técnica de detección de textura basada en el álbum de Brodatz.....	16
1.4 Metodología de desarrollo .....	17
1.5 Lenguajes.....	19
1.5.1 Lenguaje de Modelado (UML) .....	20
1.5.2 Lenguaje de Programación.....	20
1.6 Herramientas.....	21
1.6.1 Visual Paradigm.....	21
1.6.2. Geany.....	22
1.7 Object Request Broker (ORB) .....	22
Conclusiones.....	23
CAPÍTULO 2: ALGORITMOS PROPUESTOS .....	24
Introducción.....	24
2.1 Algoritmos para la detección y localización de texto.....	24
2.2 Arquitectura de los algoritmos de detección y localización.....	24

2.2.1 Conversión a escala de grises.....	25
2.2.2 Reducción de ruido.....	27
2.2.3 Detección de bordes.....	31
Conclusiones.....	39
CAPÍTULO 3: IMPLEMENTACION Y PRUEBA.....	40
Introducción.....	40
3.1 Diagrama de Componentes.....	40
3.2 Pruebas realizadas.....	43
Conclusiones.....	46
CONCLUSIONES.....	47
RECOMENDACIONES.....	48
GLOSARIO DE TÉRMINOS.....	49
BIBLIOGRAFÍA REFERENCIADA.....	51
BIBLIOGRAFÍA.....	52
ANEXOS.....	55

Fig.1 Estructura del CIPRE.....	5
Fig.2: Síntesis aditiva empleada por el modelo RGB.....	9
Fig.3: Imagen en modo RGB y en tonos grises .....	9
Fig.4: Representación de diferentes discontinuidades en la figuras .....	10
Fig.5 Máscara de Sobel .....	13
Fig.6 Máscara de Prewitt .....	13
Fig.7. a Imagen original b) Detector de contorno de Prewitt c) Detector de contorno de Sobel .....	13
Fig.8: Imagen con distintas texturas .....	14
Fig.9: Diagrama en bloque de la técnica. ....	15
Fig.10: a) Identifica zonas de diferentes texturas b) Identifica una zona de baja frecuencia.....	16
Fig.11: Diversas imágenes del álbum de Brodatz, utilizadas en el análisis de texturas. ....	16
Fig.12: Fases e Iteraciones de la Metodología RUP.....	19
Fig.13: Orden de operaciones para la detección de texto en imágenes .....	25
Fig.14: a) Imagen original RBG b) Imagen en escala de grises .....	26
Fig.15: a) Imagen Original RBG b) Imagen después de la función Convert .....	26
Fig.16: a) Imagen en escala de grises b) Imagen después del filtro Gaussiano.....	28
Fig.17: a) Imagen en escala de grises b) Imagen después del filtro Mediana .....	29
Fig.18: a) Imagen en escala de grises b) Imagen después del filtro Media .....	30
Fig.19: a) Imagen después del filtro Gaussiano b) Imagen después del operador Sobel .....	32
Fig.20: a) Imagen después del filtro Gaussiano b) Imagen después del operador Prewitt.....	33
Fig.21: Máscaras de convolución recomendadas para el obtener el filtro gaussiano.....	33
Fig.22: a) Imagen después del filtro Gaussiano b) Imagen después del operador Canny .....	36
Fig.23: a) Imagen después del filtro Gaussiano b) Imagen después del operador Robert.....	38
Fig.24: a) Imagen después del filtro Gaussiano b) Imagen después del operador ImageFilter .....	39
Fig.25: Diagrama de componentes.....	41
Fig.26 Colección de imágenes empleada. Enumeradas de 1-6 de Izq. a Der. de arriba hacia abajo....	43

## INTRODUCCIÓN

Los sistemas de prensa constituyen un medio abierto de comunicación humana, que procesa acontecimientos, ideas y sentimientos procedentes de una o varias fuentes. No son únicamente informativos, pues comparan y contraponen unos sucesos con otros, argumentan, concluyen y plantean soluciones, influyendo así en el lector. En los últimos años los sistemas de prensa han sido objeto de una constante evolución. Ello se debe en gran medida a las mejoras de las técnicas y mecanismos para su edición, impresión y distribución.

En los medios de prensa, como parte del gran sistema de información a la sociedad, se procesan a diario una gran cantidad de imágenes debido al enorme flujo de noticias que a diario circula por las redes. Las imágenes han cambiado la visión de las masas. Ellas desempeñan un rol importante en la prensa ya que estas no solo cumplen una función estética en la noticia, sino que constituyen un recurso informativo que contextualiza, amplía y apoya la información textual.

En la antigüedad la imagen era descartada muchas veces de las noticias con el objetivo de que el lector recibiera la información sin demora. Esta situación trajo como consecuencia que la lectura fuera más tediosa o que no se captara la narración de la misma. Sin embargo hoy en día es muy común encontrar en las noticias la imagen como testimonio indiscutible y transparente del acontecimiento o evento que se está reportando. La imagen, entonces, adquiere un alto valor comunicativo que cumple la función de señuelo, tanto para cautivar al lector como para llevarlo a la mejor comprensión de la narración de las noticias.

Debido al constante uso de las imágenes en la actualidad, se evidencia cada día el aumento de las mismas en los medios de prensa, por lo cual es necesario emplear diversas técnicas y tecnologías que hagan posible una búsqueda más eficiente de las mismas.

Con frecuencia es posible encontrar texto insertado en las imágenes con el fin de proporcionar información importante sobre el contenido de esta, el cual puede ser útil a la hora de realizar una búsqueda, lo que evidencia la necesidad de utilizar técnicas que permitan detectar y localizar la información que está contenida en ellas.

Los métodos para la detección y localización de texto contenido en imágenes tienen como estrategia una serie de pasos que permiten resaltar las regiones más relevantes de la imagen haciendo más sencilla la localización del texto en la misma. Una vez obtenida una imagen clara y precisa es posible detectar la información contenida en ellas.

En los medios de prensa cubanos el término búsqueda de imágenes no es empleado con frecuencia, pues es utilizado un sistema de colaboración donde los periodistas envían las noticias a publicar y la imagen escogida de este acontecimiento.

Debido a que las búsquedas de imágenes no es realizada empleando los métodos más idóneos para facilitar y agilizar este proceso, los trabajadores de los medios de prensa presentan dificultades tales como: demoras en la publicación de las noticias, así como dificultad para buscar imágenes, pues no se consulta la información contenida en las mismas, ya que no se realiza el trabajo de detectar y localizar el texto contenido en las imágenes evidenciando:

- Mal aprovechamiento de los recursos existentes.
- Lentitud de las búsquedas de imágenes.
- Tiempo real mayor al esperado al publicar las noticias.

Del análisis de la problemática expuesta anteriormente surge el **problema científico** siguiente: ¿Cómo detectar y localizar el texto contenido en imágenes para realizar una gestión más eficiente de las mismas en los sistemas de prensa cubanos?

Se define como **idea a defender** la siguiente: la implementación de algoritmos para la detección y localización de texto en imágenes facilitará en gran medida la gestión de imágenes dentro de los sistemas de prensa cubanos.

El **objeto de estudio** lo constituyen los algoritmos de detección y localización de texto en imágenes y el **campo de acción** la gestión de imágenes en los medios de prensa cubanos. Para guiar el curso de la presente investigación se plantea como **objetivo general** diseñar e implementar dos algoritmos para la detección y localización de textos en imágenes. De este se derivan los siguientes **objetivos específicos**:

- Valorar los algoritmos existentes para detección y localización de textos en imágenes.
- Seleccionar los dos métodos más eficientes y viables para la detección y localización de texto contenido en imágenes en los sistemas de prensa cubanos.
- Elegir las herramientas para el desarrollo.
- Implementar los algoritmos utilizando los dos métodos seleccionados.
- Integrar los algoritmos al producto Pyxel<sup>1</sup>.

Para darle cumplimiento a estos objetivos se han establecido las siguientes **tareas investigativas**:

- Comprender a profundidad las características propias de las imágenes digitales.

---

<sup>1</sup> **Pyxel** es un sistema web para el almacenamiento y publicación de imágenes digitales actualmente en desarrollo en el polo productivo de Gestión de la Información y el Conocimiento perteneciente a la Universidad de las Ciencias Informáticas.

- Recopilar información sobre los métodos existentes la gestión de imágenes en los medios de prensa.
- Comprobar la forma en que los medios de prensa realizan sus procesos de detección y localización de textos en imágenes.
- Examinar las herramientas libres existentes para la implementación en Python.

Los métodos científicos están constituidos por un conjunto de pasos o etapas bien establecidas que posibilitan dirigir el proceso de investigación de forma óptima, de modo que permita alcanzar el conocimiento científico, de la manera más eficiente. Es la forma organizada, sistemática y sistémica de estudiar el mundo circundante para llegar al conocimiento y comprensión de los objetos, fenómenos y procesos que lo constituyen.

Para dar cumplimiento a las tareas de investigación propuestas anteriormente, se emplearon los métodos científicos de investigación teóricos y empíricos:

Los **métodos teóricos** posibilitan estudiar las características del objeto de investigación que no son observables directamente y facilitan la construcción de modelos de investigación. Dentro de estos se emplearon:

- El **método histórico-lógico** fue útil en el análisis de la trayectoria del proceso de detección y localización de texto en imágenes para ver las principales etapas del desarrollo de los métodos existentes y de esta manera comprender la lógica interna de su funcionamiento.
- El **método analítico-sintético** empleado para procesar toda la información que se consultó en el transcurso de esta investigación y que sirvió para arribar a conclusiones en la determinación de las actividades que debían estar presentes en el desarrollo de este trabajo de diploma.
- El **método modelación** por el cual se crearon abstracciones del proceso de detección y localización de texto en imágenes con el objetivo de explicar cómo funciona dicho proceso en la realidad.
- El **método inductivo-deductivo**: permitió obtener conocimientos generalizados del proceso de detección y localización de texto en imágenes a partir del análisis de lo general a lo particular.

Por su parte los **métodos empíricos**: describen y explican las características fenomenológicas del objeto, representan un nivel de la investigación cuyo contenido procede de la experiencia y es sometido a cierta elaboración racional. Dentro de estos se utilizaron:

- El **método del experimento** mediante el cual fueron probados los métodos de detección de texto en imágenes con el objetivo de comprobar su eficiencia.

### Estructuración del contenido

El presente trabajo consta de una introducción, tres capítulos, conclusiones generales, recomendaciones, referencias bibliográficas utilizadas durante el desarrollo del mismo, glosario de términos y por último, los anexos que complementan el cuerpo del trabajo.

**Capítulo 1:** Fundamentación Teórica. Se describen conceptos importantes para el entendimiento del lector y características de los métodos de detección y localización de textos en imágenes que se deben tener en cuenta para la implementación de los mismos. Se realiza una breve descripción de la metodología y herramientas utilizadas para la implementación.

**Capítulo 2:** Algoritmos Propuestos. Se expone toda la información referente a las técnicas que se utilizan en la actualidad para la detección y localización del texto contenido en imágenes. Se proporciona una explicación detallada de la implementación de los métodos existentes.

**Capítulo 3:** Implementación y prueba. Se realiza una representación de la conexión existente entre los componentes necesarios para la detección y localización de texto en imágenes. Se documentan las pruebas realizadas a los algoritmos propuestos por el presente trabajo

## Introducción

En este capítulo se abordan conceptos de gran importancia para la comprensión del trabajo y los temas relacionados con el proceso de detección y localización de texto en imágenes. Se hace un estudio del estado del arte sobre los métodos existentes en la actualidad, tanto en el ámbito nacional como en el internacional, que aborda el procedimiento de detección de texto en imágenes. Se realiza un estudio de las principales técnicas, tecnologías, metodologías y herramientas empleadas en el desarrollo del presente trabajo.

### 1.1 Antecedentes

#### 1.1.1 Centro de Información para la Prensa

El Centro de Información para la Prensa (CIPRE), de la Unión de Periodistas de Cuba (UPEC), fue creado el 7 de febrero de 1987. Tiene como propósito coordinar la red de gestión de información y del conocimiento para el sistema de prensa cubano, cuyos productos y servicios ayudan al perfeccionamiento sistemático de los productos mediáticos cubanos.

Desde sus inicios dicha institución ha trabajado para serle útil a los periodistas, para responder a sus expectativas con el mayor nivel de eficiencia, acorde con la disponibilidad de recursos materiales y el creciente desarrollo técnico de sus recursos humanos.

Para el logro de sus objetivos actuales el CIPRE cuenta con tres grupos fundamentales de trabajo (Figura 1):

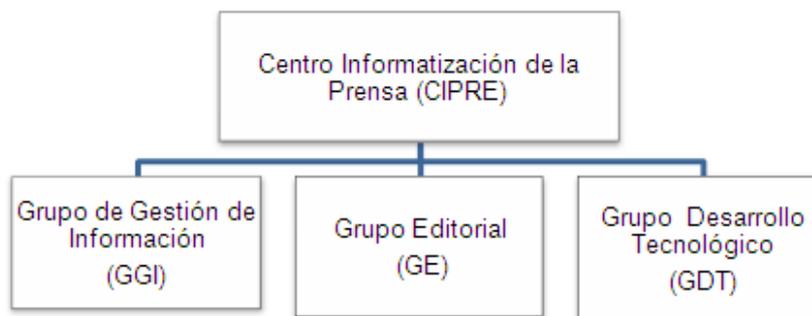


Fig.1 Estructura del CIPRE

**Grupo de Gestión de Información (GGI):** El grupo de Gestión de la Información es el encargado de la preparación de los servicios con valor agregado. Su principal objetivo es servir de base al trabajo de periodistas y comunicadores en general para generar un producto informativo final cualitativamente superior. Incluye tareas relativas a la gestión de recursos de información, procesamiento automatizado de publicaciones seriadas y no seriadas, servicios de biblioteca y hemeroteca, referencia y búsqueda de datos, elaboración de productos informativos, y marketing o estrategias de mercado en la información.

**Grupo Editorial (GE):** El Grupo Editorial tiene como responsabilidad la actualización diaria y publicación de la revista electrónica CubAhora, con trabajos exclusivos y una selección de lo mejor de la prensa en Cuba, la cual se difunde a través de INTERNET y la INTRANET del CIPRE. También asume la edición del Directorio Nacional de la Comunicación y a su vez asume otros servicios y productos informativos del CIPRE.

**Grupo Desarrollo Tecnológico (GDT):** El Grupo de Gestión de Información constituye el pilar tecnológico de toda la labor del CIPRE y su tarea primordial es la administración de la red de datos automatizados que enlaza los medios de prensa de todo el país. Gestiona, asimismo, la adquisición y puesta a punto del hardware y software necesarios para potenciar al máximo la labor técnica especializada del personal de este centro.

El GDT se encuentra inmerso en varios proyectos con el objetivo de facilitar la labor de los trabajadores de los medios de prensa cubanos. Entre estos se encuentra el proyecto de Informatización de la Prensa que es el centro de interés de la presente investigación, el cual tiene como objetivo optimizar las búsquedas de imágenes.

### 1.1.2 Ecúmene Pyxel

Actualmente en la Universidad de las Ciencias Informáticas (UCI), dentro del Polo de Gestión de la Información y el Conocimiento, existe el proyecto Informatización de la Prensa, el cual pretende constituir un sistema integrado de información a nivel de la prensa nacional, mediante la automatización de todos los procesos a través de una plataforma con estándares. Dicho sistema permitirá:

- Organizar la información existente.
- Integrar los datos.
- Recuperar el flujo de información para luego utilizarla de forma óptima.

Ecúmene Pyxel representa un sistema independiente de otros sistemas de publicación, su interfaz es abierta y es posible la construcción de sistemas que interactúen con él y aprovechen toda su

funcionalidad (**Informáticas 2008**). Este producto facilitará la labor de los trabajadores en los medios de prensa cubanos.

Pyxel brinda diversos servicios con imágenes tales como: almacenamiento a tamaño completo, catalogación y publicación con distintos perfiles<sup>2</sup> de salida. Sin embargo este sistema presenta en la actualidad la dificultad de no realizar un trabajo adecuado con las imágenes que allí se almacenan, pues no se realiza una búsqueda rápida y eficiente de las imágenes deseadas, pues no se lleva a cabo el proceso de detección y localización del texto contenido en las imágenes.

### 1.2 Conceptos

En función de darle solución a los problemas que se presentan hoy en los medios de prensa cubanos es necesario conocer disímiles conceptos que ayudarán a una mejor comprensión de lo expuesto por el presente trabajo.

#### 1.2.1 De imagen a imagen digital

El concepto de imagen surge con las representaciones pictóricas en el oriente antiguo. Con el tiempo la imagen ha ido mostrándonos los aspectos de cada cultura según los códigos de cada espacio geográfico, temporal y social. De aquí que Gubern<sup>3</sup> plantee: “En nuestra sociedad mediática las imágenes certifican la realidad y, si no hay imágenes, nada ha sucedido y nadie se inmuta”. (**Gubern 1996**)

Las imágenes han sido fundamentales en la historia de nuestra civilización ya que ha representado un sustituto afectivo, como apoyo al recuerdo y como instrumento de dominación política. Estas constituyen puntos clave para poder comprender los aspectos históricos de las sociedades de cada época ya sea en lo religioso, político, cultural, económico, artístico o sentimental.

En nuestra sociedad la concepción de la imagen ha sido transformada con el surgimiento de la imagen digital, ya que esta adquirió, gracias a las tecnologías, cierta independencia: pueden ser alteradas sin dañarse físicamente, es posible hacerle modificaciones sin perder la original y la forma de poseerlas también es distinta, ya que se guardan en la memoria de la computadora o cualquier otro medio de almacenamiento sin necesidad de cobrar forma material en una hoja de papel.

---

<sup>2</sup> **Perfiles:** el modo que tiene Pyxel de entregar una imagen más apropiada para el medio donde será finalmente publicada.

<sup>3</sup> **Román Gubern:** Historiador y teórico de la comunicación audiovisual. Catedrático en la Universidad Autónoma de Barcelona. Se especializa en el análisis de lo que él denomina la “ionósfera contemporánea”: un medio ambiente de imágenes inmersivas que modifican nuestra percepción, nuestra conducta, nuestra vida cotidiana.

### 1.2.2 Imagen digital

Una imagen digital<sup>4</sup> es un arreglo bidimensional de píxeles<sup>5</sup>. El valor de cada píxel se representa mediante una función  $f$ , donde  $f(x,y)$  representa el nivel de brillantez, color o intensidad de la imagen en tales coordenadas. Por lo tanto, una imagen en blanco y negro puede ser representada por una matriz de dimensión  $m \times n$ , donde  $f(x,y) \in \{0,1\}$ . Además de la representación en blanco y negro, otras opciones son las imágenes en tonos grises, para las cuales  $f(x,y)$  representa un nivel de intensidad típicamente asociando a un entero entre 0 y 255 y otras más sirven para describir imágenes en color.

Una imagen digital, puede ser también definida como una función en dos dimensiones,  $f(x,y)$ , donde:  $x$  e  $y$  son coordenadas espaciales, la amplitud de  $f$  en el punto  $(x,y)$  es conocida como la intensidad de la imagen en ese punto. Tanto, los valores de las coordenadas  $x$  e  $y$ , como de la amplitud de la función  $f$  son valores finitos. **(Rodríguez 2003)**

En general una imagen digital es cualquier imagen fija o en movimiento que se capture en un medio electrónico y que se represente como un archivo de información leído como una serie de pulsos eléctricos.

### 1.2.3 Imagen digital a colores

Actualmente existen varias representaciones de imágenes de color entre las que se encuentran: RGB, CMKY, HSV, HSI, entre otras. La diversidad de espacios o representaciones del color es explicada por todos los caminos que ha seguido el hombre en aras de esclarecer el proceso de intensidad de tres componentes, Rojo, Verde y Azul (o RGB por sus siglas en inglés: Red, Green, Blue).

RGB se basa en el modelo de síntesis aditiva, con el que es posible representar un color mediante la mezcla por adición de los tres colores luz primarios. Este modelo no define por sí mismo lo que significa exactamente rojo, verde o azul, por lo que los mismos valores RGB pueden mostrar colores notablemente diferentes en diferentes dispositivos que usen este modelo de color.

---

<sup>4</sup> La definición de "**digital**" es: toda información representada por una serie de pulsos eléctricos basados en un sistema binario (ceros y unos).

<sup>5</sup> El **píxel** es la unidad más pequeña y diminuta de una imagen digital y está presente en un inmensurable número para formar una imagen completa.



Fig.2: Síntesis aditiva empleada por el modelo RGB

### 1.2.4 Imagen digital en escala de grises

Es una matriz de  $M \times N$  elementos numéricos, siendo este número la intensidad luminosa en un determinado punto o píxel. Esta es una escala empleada en la imagen digital en las que el valor de cada píxel posee un valor equivalente a una graduación de gris. Las imágenes representadas de este tipo están compuestas de sombras de grises, que van desde el negro más profundo, variando gradualmente en intensidad de grises hasta llegar al blanco.



Fig.3: Imagen en modo RGB y en tonos grises

### 1.2.5 Detección y localización de texto en imágenes

Con frecuencia es posible encontrar texto insertado en las imágenes, ya sea nombres de personas, títulos, fechas de eventos entre otras. Esta situación plantea una necesidad básica: la búsqueda y recuperación de contenidos en las imágenes deseadas. El método tradicional de recuperación, que es la detección manual del texto contenido en la imagen, se muestra insuficiente e inadecuado por diversos motivos. Básicamente es posible destacar dos problemas:

- La cantidad de trabajo que se requiere para obtener informaciones contenidas en las imágenes y expresar el rico contenido presente en ellas es inmenso.

- Resulta tedioso y difícil recuperar las informaciones contenidas en las imágenes, además de estar sujeto a errores humanos.

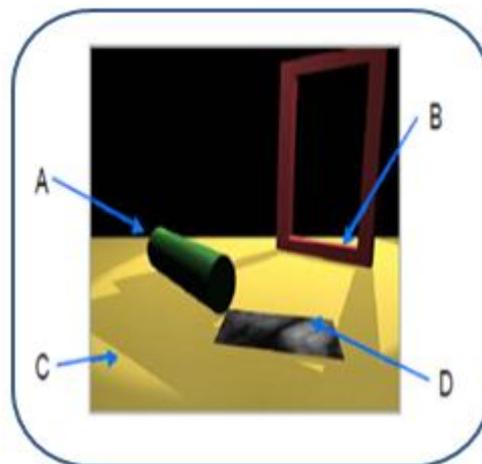
Para solucionar estos problemas surgieron unas series de técnicas o métodos que se pueden aplicar para la recuperación de texto en las imágenes de manera eficiente.

### 1.3 Métodos de detección de textos en imágenes

La detección de texto en imágenes es un campo de investigación relativamente joven y en constante desarrollo. No es hasta hace algunos años que se supera la barrera de complejidad computacional que supone la manipulación de archivos de imágenes, propiciando esto la proliferación de novedosas técnicas en este marco tecnológico. Actualmente se pueden identificar dos tendencias fundamentales en las técnicas o métodos vinculados a la detección de texto en imágenes: los basados en la detección de bordes y los basados en análisis de textura.

#### 1.3.1 Métodos basados en la detección de bordes

Se define como detección de bordes o detección de contornos, en términos de procesado digital de imágenes, la detección de aquellas zonas en las que se produce un fuerte cambio de intensidad. Los bordes son frecuentemente relacionados con discontinuidades dentro de un conjunto de píxeles. El contorno de imagen digital corresponde al mínimo conjunto de píxeles que separa ese objeto del fondo o background de la imagen. Normalmente estos contornos se corresponden con los puntos donde se producen discontinuidades en los valores de píxeles adyacentes o con los puntos donde cambia un patrón que se repite es decir cambios de textura.



**Fig.4: Representación de diferentes discontinuidades en la figuras**

**A: Discontinuidad de profundidad: cambios abruptos de profundidad en la escena**

**B: Discontinuidades normales de superficies: cambios en la orientación de la superficie**

**C: Discontinuidades de iluminación: sombras, cambios de luz**

**D: Discontinuidades de reflectancia: propias de las superficies, marca huellas**

Cuando se habla de detección de bordes se piensa en una aplicación que dé como resultado un contorno, sin embargo el real objetivo de estos métodos es obtener imágenes que a su salida muestren píxeles de mayor intensidad en los valores que detecten transiciones cercanas.

Los métodos de detección de bordes utilizan diversos operadores. Entre ellos es posible destacar dos ramas:

- Técnicas basadas en cruces por cero
  - Detector de Canny
- Técnicas basadas en el gradiente<sup>6</sup>
  - Operador de Sobel
  - Operador de Prewitt
  - Operador de Roberts

La detección de los bordes de una imagen es de suma importancia y utilidad, pues facilita muchas tareas, entre ellas, el reconocimiento de objetos, la segmentación de regiones, entre otras. También resulta útil en diversas disciplinas, como cartografía o medicina, cuyo se pretende preservar importantes propiedades estructurales o cuyo se plantea la necesidad de distinguir formas o reconocer figuras dentro de una imagen.

### 1.3.1.1 Detector de Canny

El algoritmo de Canny es usado para detectar todos los bordes existentes en una imagen. Este algoritmo está considerado como uno de los mejores métodos de detección de contornos mediante el empleo de máscaras de convolución y basado en la primera derivada. Esta última toma el valor de cero en todas las regiones donde no varía la intensidad y tiene un valor constante en toda la transición de intensidad. Por tanto un cambio de intensidad se manifiesta como un cambio brusco en la primera derivada, característica que es usada para detectar un borde.

Los puntos de contorno son zonas de píxeles en las que existe un cambio brusco de nivel de gris. En el tratamiento de imágenes, se trabaja con píxeles, y en un ambiente discreto, es así que en el algoritmo de Canny se utiliza máscaras, las cuales representan aproximaciones en diferencias finitas.

#### Criterios del Algoritmo de Canny

---

<sup>6</sup> El **gradiente** es un vector que se encuentra normal a una superficie o curva en el espacio a la cual se le está estudiando, en un punto cualquiera. Mide la tasa y la dirección del cambio en un campo escalar; el gradiente de un campo escalar es un campo vectorial.

En 1986, Canny propuso un método para la detección de bordes, el cual se basaba en tres criterios, estos son:

- El criterio de detección expresa el hecho de evitar la eliminación de bordes importantes y no suministrar falsos bordes.
- El criterio de localización establece que la distancia entre la posición real y la localizada del borde se debe minimizar.
- El criterio de una respuesta que integre las respuestas múltiples correspondientes a un único borde.

El algoritmo de Canny es guiado por los siguientes pasos tomados de Detección de bordes mediante el algoritmo de Canny. **(Rebaza)**

1. Obtención del gradiente: Donde se calcula la magnitud y orientación del vector gradiente en cada píxel.
2. Supresión no máxima: Se logra el adelgazamiento del ancho de los bordes, obtenidos con el gradiente, hasta lograr bordes de un píxel de ancho.
3. Histéresis de umbral: Se aplica una función de histéresis basada en dos umbrales; con este proceso se pretende reducir la posibilidad de aparición de contornos falsos.
4. Cierre de contornos abiertos: Se utiliza para cerrar los contornos que pudiesen quedar abiertos por problemas de ruido.

### 1.3.1.2 Operador de Sobel y operador de Prewitt

El operador Sobel es usado en el procesamiento de imágenes, para la detección de bordes fundamentalmente. Este es un operador diferencial que calcula una aproximación al gradiente de la función intensidad de una imagen.

Representa una primera aproximación imprecisa del gradiente de la imagen, pero es de calidad suficiente para ser de uso práctico en muchas aplicaciones. Más precisamente, éste operador utiliza sólo valores de intensidad en una región de 3x3 alrededor de cada punto analizado para calcular el gradiente correspondiente, además de que utiliza sólo números enteros para los coeficientes que indican la aproximación del gradiente.

-1	0	1	-1	-2	-1
-2	0	2	0	0	0
-1	0	1	1	2	1
$S_x$			$S_y$		

**Fig.5 Máscara de Sobel**

Prewitt al igual que Sobel expande la definición del gradiente en una máscara de 3x3 para reducir el ruido, utiliza la misma ecuación que el operador anterior pero con una constante  $c=1$ . A continuación se muestra la máscara utilizada por este.

-1	0	1	-1	-1	-1
-1	0	1	0	0	0
-1	0	1	1	1	1
$S_x$			$S_y$		

**Fig.6 Máscara de Prewitt**

Las siguientes imágenes son el resultado de aplicar a una imagen los detectores de contorno de Sobel y Prewitt.



**Fig.7. a Imagen original    b) Detector de contorno de Prewitt    c) Detector de contorno de Sobel**

### 1.3.1.3 Operador Roberts

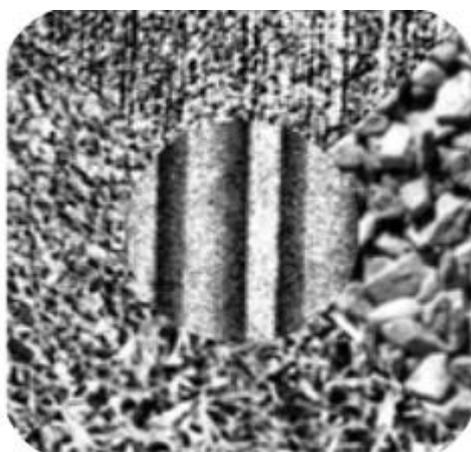
En el año 1965 fue introducido el primer operador para la detección de bordes entre las diferentes partes de una imagen, este fue nombrado operador de Roberts. Fue el primer paso en el proceso de descomponer una imagen en sus diversos elementos. Desde entonces surgieron un gran número de técnicas y algoritmos encaminados a la desarrollar el campo de los procesos de segmentación de una imagen.

Roberts obtiene una buena respuesta ante bordes diagonales y ofrece buenas prestaciones en cuanto a localización. Este tiene como gran inconveniente su extremada sensibilidad al ruido, por lo cual tiene pobres cualidades de detección.

### 1.3.2 Métodos basados en la detección de textura

Intuitivamente la textura de un objeto dentro de una imagen es el conjunto de formas que se aprecia sobre su superficie y que lo dota de cierto grado de regularidad. Entre las diversas definiciones de textura se encuentra que es uno o más patrones los cuales se repiten de manera periódica.

Los métodos de detección basados en el análisis de textura manipulan el texto de la imagen como un tipo de textura bien definida y que es ampliamente diferenciable del resto de la imagen donde está insertado. El análisis de detección de textura emplea tres técnicas para el procesamiento digital de imágenes, estas son: estadística, estructural y espectral. La estadística es empleada para indicar si una imagen es suave o burda, por otro lado la estructural indica qué arreglos primitivos existen en la imagen tales como la regularidad de líneas paralelas, mientras que la técnica espectral está basada en las propiedades de Fourier y es principalmente usada para detectar periodicidad global en una imagen identificando picos de alta energía en su espectro. Estas técnicas pueden ser usadas por separado o pueden apoyarse unas a otras con el objetivo de detectar diferentes texturas.



**Fig.8: Imagen con distintas texturas**

El método propuesto por Jorge O. Lugo(Lugo), basado en el análisis de multiresolución, propone tratar las zonas de distintas frecuencias espaciales de la imagen como lugares con texturas diferentes.

El análisis de multiresolución permite estudiar una imagen en distintas resoluciones y extraer características propias de cada resolución, el mismo fue realizado a través de la transformada wavelet, ya que en la definición de la misma es posible apreciar, a partir de la constante de dilatación  $a$ , el

análisis de multiresolución se encuentra en forma implícita, donde  $Y(x)$  es la wavelet madre. Los valores de las transformadas son funciones de dos parámetros  $a$  y  $b$ ; el parámetro  $b$  indica el lugar de donde proviene el valor transformado y el parámetro  $a$  es el factor de escala que fija el nivel de resolución en el análisis de la imagen.

$$w(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(x) \Psi\left(\frac{x-b}{a}\right) dx$$

Los  $w(a, b)$  son también llamados coeficientes de detalles, cada uno de estos coeficientes indican los detalles o frecuencias que puedo hallar en la imagen y los lugares de donde provienen estos detalles o frecuencias. Para poder caracterizar una textura se consideran sus frecuencias espaciales. Si se desea detectar o contornear zonas de diferentes tipos de texturas debe ser realizado con valores de  $a < 1$  lo más pequeño posible para buscar las altas frecuencias en la imagen y si se desea localizar zonas de bajas frecuencias se debe aumentar el valor de  $a$ . Para detectar texturas diferentes se aplica la transformada de wavelet.

Los métodos basados en la detección de textura no son muy utilizados en la actualidad, esto se debe a la inexistencia de bibliografía que demuestre que estos métodos son eficientes. La mayor dificultad de esta técnica radica en que realiza la detección de textura de manera muy inexacta.

### 1.3.21 Técnica de detección de textura basada en la transformada wavelet

La técnica para detectar diferentes texturas es la siguiente, primero se realiza la transformada wavelet en una resolución determinada, según que frecuencias se busquen asociadas con esas texturas en la imagen; luego se aplica un filtro pasa altos a estos coeficientes y se multiplica por un factor  $\alpha$  a los que quedan para que al realizar la transformada wavelet inversa resalte los detalles en forma más pronunciada, y por último, es restada la imagen original de la filtrada, de este modo se obtienen los puntos en la imagen que dan la separación de las diferentes texturas o zonas de igual frecuencia.

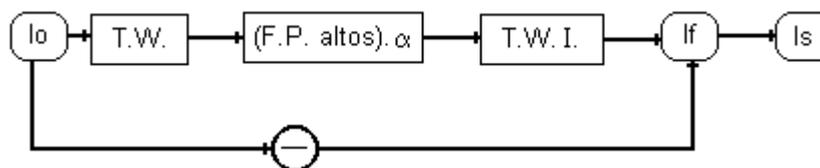


Fig.9: Diagrama en bloque de la técnica.

Io: imagen original

If: imagen filtrada

Is: imagen de salida.

Para realizar las transformadas se emplean las wavelets Haar y lineal. En las figuras 11 y 12 se aprecia el resultado del diagrama mostrado en la Fig. 10

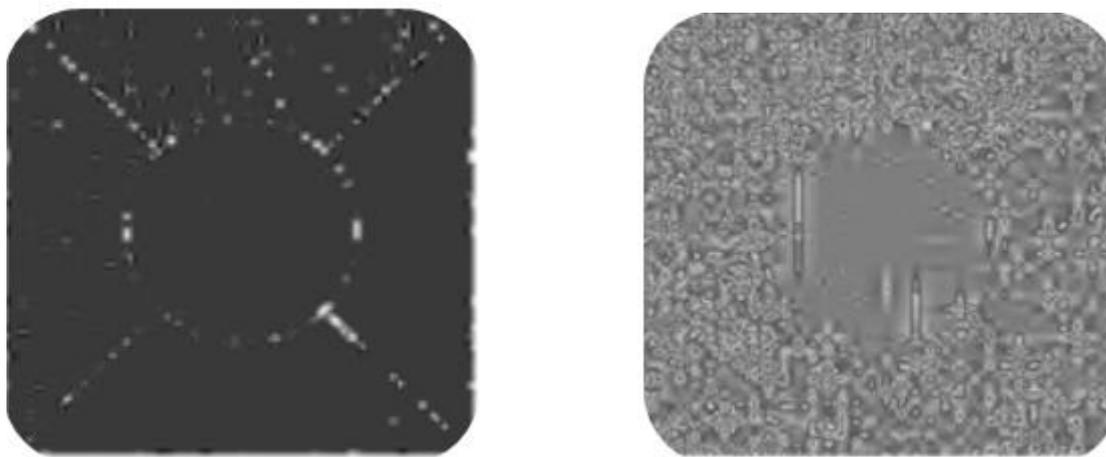


Fig.10: a) Identifica zonas de diferentes texturas      b) Identifica una zona de baja frecuencia

### 1.3.2.2 Técnica de detección de textura basada en el álbum de Brodatz

Para el estudio y comparación de algoritmos sobre imágenes que presentan texturas suelen utilizarse como referencia las imágenes de Brodatz, conocidas como álbum de Brodatz<sup>7</sup>. Este álbum contiene 154 imágenes. En la siguiente figura se muestra algunas imágenes de este álbum.

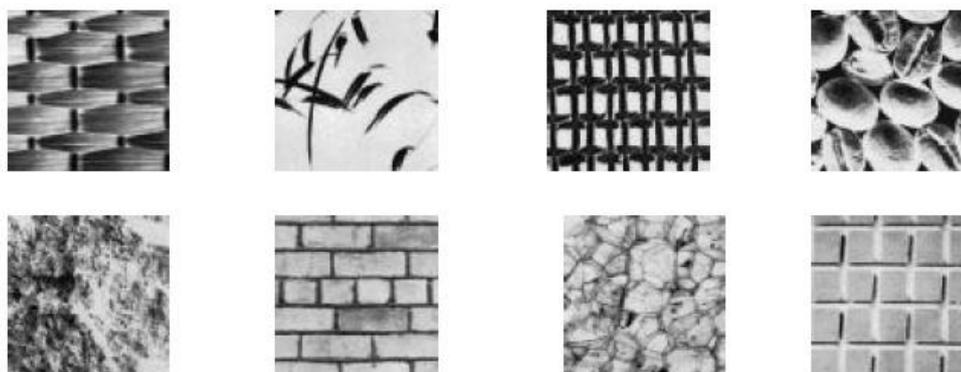


Fig.11: Diversas imágenes del álbum de Brodatz, utilizadas en el análisis de texturas.

Existen dos enfoques para definir una textura: uno descendente (“topdown”) y otro ascendente (“bottom-up”). El enfoque descendente se basa en la existencia de un elemento básico de textura,

<sup>7</sup> album de Brodatz (P. Brodatz, "Textures: A Photographic Album for Artists and Designers", Dover Publications, New York, 1966)

llamado texel, y en una regla de formación. Esta regla define cómo y dónde se sitúan estos elementos básicos. Este enfoque funciona bien cuando la textura es bastante regular, por ejemplo en la imagen de una pared de ladrillos. Por otro lado, el enfoque ascendente se basa en que la textura es una propiedad que se puede derivar de estadísticos (como la media y la varianza) de pequeños grupos de píxeles. Este enfoque funciona bien para texturas donde resulta difícil ver los componentes individuales. No obstante, la línea divisoria entre los dos enfoques no es clara.

### **1.3.3 Detectores de texto seleccionados**

Para la realización del presente trabajo fueron seleccionados los detectores de contornos basados en bordes. Estos métodos son seguros y eficientes, además están respaldados por diversos investigadores que afirman la veracidad de los mismos. Cuentan con una gran variedad de bibliografía que hace posible un mejor estudio y comprensión de los mismos.

### **1.4 Metodología de desarrollo**

El desarrollo de un software con calidad depende de varias actividades regidas por etapas, donde la elección de la mejor metodología a usar para un determinado proyecto es trascendental para el éxito del producto a desarrollar.

Actualmente no existe una metodología de software universal. Las características de cada proyecto, ya sea de recursos, de tamaño del proyecto o del equipo de desarrollo, exigen que el proceso sea configurable y determina la metodología necesaria para su elaboración. El desarrollo de software no es una tarea sencilla y como prueba de esto es la inmensidad de propuestas metodológicas existentes. Se han destacado dos corrientes fundamentales de las mismas, por una parte se encuentran aquellas propuestas más tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán y por otro las ligeras que tratan de mejorar la calidad del software a través de una comunicación directa e inmediata entre las personas que intervienen en el proceso.

Muchas veces no se toma en cuenta la utilización de una metodología adecuada. En ocasiones, ocurre que se realiza el diseño del software de manera rígida con los requerimientos que el cliente solicitó, de tal manera que cuando el cliente en la etapa final o de prueba, solicita un cambio se hace muy difícil realizarlo, pues si se hace, altera muchas cosas que no se han previsto, y es justo éste uno de los

factores que ocasiona un atraso en el proyecto y por tanto la incomodidad del desarrollador por no cumplir con el cambio solicitado y el malestar por parte del cliente por no tomar en cuenta su pedido. Para evitar estos incidentes se debe llegar a un acuerdo formal con el cliente, al inicio del proyecto, de tal manera que cada cambio o modificación no perjudique el desarrollo del mismo, así como escoger una metodología adecuada para el desarrollo del software que sirva como guía para realizar de forma disciplinada y eficiente el producto deseado.

### 1.4.1 Metodología seleccionada

El Proceso Racional Unificado RUP(Rational Unified Process), constituye la metodología estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos. Proporciona una aproximación disciplinada a la asignación de tareas y responsabilidades. RUP actúa como modelo y puede ser adaptado y extendido(Jacobson y Booch 2004). Esta metodología ha sido empleada en la elaboración del producto Ecúmene Pyxel desde sus inicios, por lo cual ha sido la seleccionada para la elaboración del presente trabajo, además porque es un proceso de ingeniería de software que permite producir software flexible y de calidad en plazos y presupuestos predecibles.

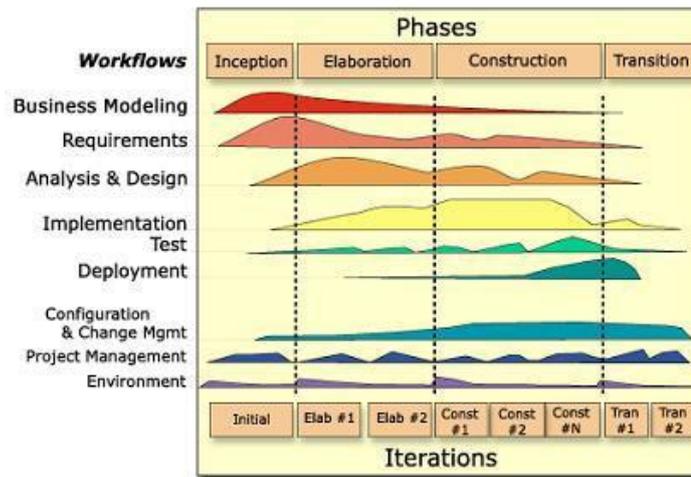
Características esenciales de RUP:

- Dirigido por los Casos de Uso: Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, estos representan los requisitos funcionales
- Centrado en la arquitectura: La arquitectura muestra una visión común del sistema completo en la que el equipo de trabajo y los usuarios deben de estar de acuerdo, por lo que describe los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente.
- Iterativo e incremental: Propone que cada fase se desarrolle en iteraciones. Una iteración implica actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros.

RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades.

- Inicio: Las iteraciones hacen mayor énfasis en actividades del modelado de negocio y requerimiento.
- Elaboración: Las iteraciones se orientan al desarrollo de la arquitectura, abarcan más los flujos de trabajo de requerimientos, modelo de negocios (refinamiento), análisis, diseño y una parte de la implementación orientada a la arquitectura.

- **Construcción:** Se lleva a cabo la construcción del producto por medio de una serie de iteraciones.
- **Transición:** Se pretende garantizar que se tiene un producto preparado para su entrega a la comunidad de usuarios.



**Fig.12: Fases e Iteraciones de la Metodología RUP**

RUP usa el lenguaje UML en la preparación de todos los planos del sistema. Este es una parte integral del Proceso Unificado, ya que se desarrolla a la par y utiliza el paradigma orientado a objeto para su descripción.

## 1.5 Lenguajes

Un lenguaje de programación es un idioma artificial que permite crear programas y software. Están conformados por diversos símbolos y reglas que definen su estructura y el significado de los elementos que lo integran. Existen diversos lenguajes de programación entre los que se encuentran: Delphi, Visual Basic, Pascal, Java y Python.

Por otra parte está el lenguaje de modelado, el cual es descrito como la notación, principalmente gráfica, que usan los métodos para expresar un diseño. De estos es posible identificar tres generaciones de técnicas:

- En la primera generación, tecnólogos aislados y grupos pequeños desarrollaban técnicas que resolvían problemas que se encontraban de primera mano en los proyectos de desarrollo orientado a objetos. En esta generación se incluye a autores y técnicas como Rumbaugh, Jacobson, Booch, entre otros.
- En la segunda generación se reconoció que muchas de las mejores prácticas pertenecían a diferentes métodos del fragmentado terreno de la metodología orientada a objetos.

- La tercera generación consiste en intentos creíbles de crear dicho lenguaje unificado por la industria, cuyo mejor ejemplo es UML

### 1.5.1 Lenguaje de Modelado (UML)

El ser humano requiere de modelos para manejar sistemas complejos, y en cuanto más complejos se vuelven los sistemas, es necesario tener mejores técnicas de modelado. El contar con una metodología universal para el desarrollo de sistemas de software es de gran beneficio en la construcción de todo tipo de sistemas. El lenguaje unificado de modelado (UML: *Unified Modeling Language*) es un lenguaje para especificar, construir, visualizar y documentar los artefactos de un sistema de software Orientado a Objetos (OO). Ofrece varias ventajas:

- Es un lenguaje fácil de aprender y su código es reutilizable.
- Ofrece una serie de diagramas mediante los cuales es posible modelar sistemas. Entre ellos se encuentran: Diagrama de Casos de Uso, Diagramas de Componentes, etc.
- Es posible modelar diversos tipos de sistemas ya sean de software, de hardware o de organizaciones del mundo real.

### 1.5.2 Lenguaje de Programación

Python constituye un lenguaje de programación multiparadigma pues permite utilizar varios estilos de programación tales como: programación orientada a objetos, programación estructurada y programación funcional.

Entre las características fundamentales por las que Python fue seleccionado para la implementación del presente trabajo se encuentran:

- Es limpio y elegante para programar pues permite fragmentar el programa en módulos reutilizables desde otros programas y ahorra un tiempo considerable en el desarrollo del programa, ya que no es necesario compilar ni enlazar, el intérprete se puede utilizar de modo interactivo, facilitando experimentar con características del lenguaje, escribir programas desechables o probar funciones durante el progreso del programa.
- Puede utilizarse como un lenguaje de extensión para módulos y aplicaciones que necesitan de una interfaz programable
- Permite dividir el código en diferentes módulos que se pueden reutilizar en otros programas implementados en él simplemente importándolos.
- Cuenta con diferentes módulos estándar que proporcionan multitud de aplicaciones para utilizar de base para la creación de programas. Existen además una gran variedad de módulos

adicionales que pueden ser descargados y utilizados fácilmente. De estos módulos ha sido utilizado para la realización de la presente aplicación: PIL (Python Library Imaging) la cual es la encargada del procesamiento de imágenes.

Python es sin dudas un lenguaje de muy alto nivel y que por lo tanto incluye tipos de datos muy complejos que permiten expresar operaciones complejas con sentencias relativamente simples, es además un lenguaje muy sencillo, compacto y legible.

### 1.6 Herramientas

Las herramientas CASE de sus siglas en inglés Computer Aided Software Engineering en español Ingeniería de Software Asistida por Ordenador, son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y dinero. Estas herramientas tienen como objetivos:

- Mejorar la productividad en el desarrollo y mantenimiento del software.
- Aumentar la calidad del software.
- Mejorar el tiempo, coste de desarrollo y mantenimiento de los sistemas informáticos.
- Mejorar la planificación de un proyecto.
- Aumentar la biblioteca de conocimiento informático de una empresa ayudando a la búsqueda de soluciones para los requisitos.

Dentro de la lista de aplicaciones CASE se encuentra Visual Paradigm para UML, que cuenta con una versión gratuita denominada Community Edition, traducida al español Edición Comunitaria.

Por otro lado las herramientas de entorno de desarrollo integrado (IDE: Integrated Development Environment) son un conjunto de programas que se ejecutan a partir de una única interfaz de usuario. De estas fue seleccionada Geany por ligero y sencillo. Esta aplicación multiplataforma permite trabajar con los archivos fuente de multitud de lenguajes de programación: Java, Javascript, C, C++, C#, HTML, XML, Pascal, Fortran, Python, etc. Además es bastante rápido si es comparado con otras soluciones más complejas como Eclipse o NetBeans, gracias a que mantiene muy pocas dependencias de otros paquetes y una extremadamente sencilla interfaz.

#### 1.6.1 Visual Paradigm

Visual Paradigm para UML es una de las herramientas CASE, que proporciona excelentes facilidades de interoperabilidad con otras aplicaciones, considerada como: muy completa y fácil de usar y con soporte multiplataforma.

Visual Paradigm está diseñado para una amplia gama de usuarios, incluidos los Ingenieros de Software, Analistas de Sistemas, Analistas de Negocios, y para aquellas personas interesadas en la construcción de sistemas de software de forma fiable a través de la utilización del enfoque Orientado a Objetos.

Esta herramienta es muy fácil de instalar y actualizar. Además permite invertir código fuente de programas, archivos ejecutables y binarios en modelos UML al instante, creando de manera simple toda la documentación. Incorpora el soporte para trabajo en equipo, que permite que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios realizados por sus compañeros.

### 1.6.2 Geany

Geany es un pequeño y ligero editor ligero editor de textos para programadores y desarrolladores web disponibles tanto para Windows como para Linux. Dispone de las funciones básicas de un editor, siendo necesario para su uso tener instalados la librería: gtk runtime. Presenta características básicas del entorno de desarrollo integrado (IDE). Es multiplataforma ya que puede ser ejecutado en varios sistemas operativos. Soporta varios lenguajes web tales como C, C++, Java, JavaScript, PHP, HTML, CSS y Python. Dentro de sus principales características están: Soporte multi-documento, autocompletado y soporte de proyectos. Este software es distribuido bajo licencia GNU GPL como software libre.

### 1.7 Object Request Broker (ORB)

Un ORB es una capa de software que permite a los objetos realizar llamadas a métodos situados en máquinas remotas utilizando la red. Esta maneja la transferencia de datos utilizando un estándar que le permita convertir las estructuras de los datos de manera que sean compatibles entre los dos objetos, proceso denominado marshalling (clasificación).

Esta permite a los objetos que se encuentran distribuidos en diversas máquinas interactuar como si estuviesen en una sola. ORB soporta cuatro tipos de interfaces de objetos:

- Servicio de Objetos(OS: Object Service): Interfaces para servicios generales. Usados en cualquier programa basados en objetos distribuidos.
- Facilidades Comunes (CF: Common Facilities): Interfaces orientadas al usuario final programadas por la aplicación.
- Interfaz de Dominio (DF: Domain Interface): interfaces de dominio específico para las aplicaciones

- Interfaz de Aplicación (AI: Application Interfaces): Este tipo de interfaz acepta interfaces que no sean estandarizadas y se utilizan en aplicaciones específicas.

Existen diversas tecnologías basadas en ORB. Entre ellas se encuentran:

- RMI de Java (Remote Method Invocation)
- CORBA (Common Object Resource Broker Architecture)
- XML-RPC (XML Remote Procedure Call)
- Pyro (Python Remote Object)

Pyro (Python Remote Object).

Entre los diversos tipos ORB se encuentra Pyro (Python Remote Object). Este es un avanzado y potente sistema de tecnología de objetos distribuidos implementado en Python, el cual posee una librería sencilla y potente que simplifica el uso de objetos compartidos en la red. Pyro es una librería sencilla y potente que simplifica el uso de objetos compartidos en la red y que realiza la comunicación entre componentes mediante el envío de mensajes. Es mucho más simple de usar en comparación con las otras tecnologías mencionadas.

### **Conclusiones**

En la primera parte de este capítulo se definieron los conceptos relacionados con el objeto de estudio definido en la introducción del documento. En la segunda parte, se desarrolló el estudio de algunos de los métodos actuales que fueron empleados durante el desarrollo del software, mencionando sus características y ventajas; se definió la utilización de herramientas con soporte multiplataforma y licencias de utilización libre. Se abordó sobre las metodologías, lenguaje de modelado, lenguaje de programación y las herramientas a utilizar.

## **Introducción**

En este capítulo se analizan los métodos de detección y localización de texto en imágenes. Es explicado el orden lógico que estos algoritmos deben seguir. Se describe el funcionamiento de cada una de las técnicas empleadas para la realización de este procedimiento. Finalmente se realiza la selección de los métodos de detección de contorno a emplear en esta investigación basada en el análisis de resultados experimentales.

### **2.1 Algoritmos para la detección y localización de texto**

Los métodos de detección de texto que este trabajo propone se basan en la detección de bordes horizontales y verticales de una imagen. Estos métodos reducen de manera significativa la cantidad de información que contienen las imágenes y filtran todos aquellos datos irrelevantes para la detección del texto, preservando las características principales de la imagen, es por esto que constituyen una herramienta poderosa en el procesamiento de las mismas. Son significativos en varios aspectos como por ejemplo: robustos en la detección de texto de disímiles colores y tamaños, además son de muy bajo costo computacional y temporal.

### **2.2 Arquitectura de los algoritmos de detección y localización**

Existen varias maneras para realizar la detección de texto en imágenes. Una de ellas es aplicando varios algoritmos para detectar los contornos y así resaltar las regiones que contengan textos. Otro es auxiliándose de una función de la librería PIL para filtrar la imagen y así obtener resaltado el texto. En esta última manera de detección el usuario debe cargar la imagen a la cual le quiere realizar la operación.

Los algoritmos propuestos por este trabajo reciben como entrada una imagen la cual es brindada por el usuario, a partir de ella se detecta el texto contenido en la misma. Para detectar el texto es necesario aplicar diversas operaciones en un debido orden. El primer paso es verificar si la imagen es a color o no, en caso positivo esta es transformada a escala de grises. Posteriormente se aplica un filtro para reducir el posible ruido presente en la imagen y finalmente es utilizado un detector de contorno para conseguir la imagen con el texto resaltado.

El mecanismo actual presenta como estructura la mostrada en la Fig. 14. Para esta estructuración se identifican los bloques que lo constituyen.



Fig.13: Orden de operaciones para la detección de texto en imágenes

### 2.2.1 Conversión a escala de grises

La detección de las posibles regiones del texto es el primer paso para la detección del texto propuesto. El proceso de conversión de una imagen a color a escala de grises es realizado de diversas maneras, una de ellas es aplicando una sencilla transformación matemática. Se verifica si el modo de la imagen es RGB, en caso positivo, mediante la utilización de diversas funciones de la librería PIL, se obtienen los valores de los píxeles. A estos valores se le aplican la siguiente función matemática:

$$\text{Transformación} = R + B + G/3$$

Teniendo en cuenta que el valor resultante debe encontrarse en el rango que la escala de grises proporciona (de 0 a 256).

A continuación se muestra la imagen original utilizada para realizar las pruebas de los algoritmos propuestos y la imagen resultante luego de aplicarle el algoritmo diseñado:



Fig.14: a) Imagen original RBG

b) Imagen en escala de grises

Otra forma de realizar este procedimiento es utilizando una función de la librería PIL llamada *convert*. A esta se le debe insertar el modo de la imagen para que realice la conversión. En caso de que este no le sea proporcionado, selecciona un modo de forma automática. Mientras realiza este proceso se auxilia de la biblioteca *UIT-R 601-2 luma* utilizando la siguiente función:

$$L = R * 299/1000 + G * 587/1000 + B * 114/1000$$

La imagen resultante de este algoritmo es la siguiente:



Fig.15: a) Imagen Original RBG

b) Imagen después de la función Convert

Una vez obtenida la imagen en escala de grises es necesario aplicarle un filtro con el objetivo de eliminar los posibles ruidos presentes en la ella y de esta manera suavizar o realzar los detalles más significativos de la misma.

### 2.2.2 Reducción de ruido

Las técnicas de reducción de ruido consisten en resaltar o suprimir, de forma selectiva, información contenida en una imagen a diferentes escalas espaciales, para destacar algunos elementos de esta o para ocultar valores anómalos de ella.

El ruido es cualquier entidad presente en las imágenes, es decir, datos o resultados intermedios que no son importantes para la computación que se desea realizar.

Básicamente, hay tres tipos diferentes de filtros para reducir el ruido:

- Filtros de paso bajo:  
Este deja inalterables las bajas frecuencias y atenúa o elimina las altas frecuencias. El resultado en el dominio del espacio consiste en un suavizado: eliminar pequeños detalles y ruidos de la imagen.
- Filtros de paso alto:  
Deja inalterables las altas frecuencias y atenúa o elimina las bajas frecuencias. El resultado en el dominio del espacio consiste en un realzado de los cambios bruscos de niveles de grises. De ahí que se use para detectar bordes.
- Filtros de banda:  
Un filtro de banda atenúa las altas y bajas frecuencias, pero mantiene intactas las frecuencias que se encuentren en una banda determinada.

Para realizar la reducción de ruido presente en la imagen existen varios métodos.

#### Filtro Gaussiano:

El filtro gaussiano es capaz de implementar diversas máscaras que imitan la forma de una expresión gaussiana:

$$G(x,y) = e^{- (x + y)^2 / 2\sigma^2}$$

Donde  $x$ ,  $y$  representan las coordenadas de la imagen mientras que  $\sigma$  es la desviación estándar de la probabilidad de distribución asociada. La desviación estándar  $\sigma$  es el único parámetro del filtro gaussiano y es proporcional al tamaño del área de los vecinos más cercanos en que opera este filtro. Los píxeles más lejanos del centro del operador tienen menor influencia y los píxeles más allá de tres valores de  $\sigma$  a partir del centro tienen una influencia insignificante. En la siguiente figura se muestran los resultados luego de aplicar este procedimiento.



Fig.16: a) Imagen en escala de grises      b) Imagen después del filtro Gaussiano

El filtro gaussiano utilizado tiene como inconveniente que, al remover el ruido contenido en la imagen causa el empañamiento de la misma ocasionando pérdida de los detalles más finos de esta. Este es frecuentemente empleado en aplicaciones para la detección de bordes y en el análisis de escala espacial.

### Filtro Mediana

Otro de los filtros empleados con frecuencia para reducir el ruido en las imágenes es el filtro mediana. La mediana de un determinado número de valores es tal que deja la mitad de los valores a un lado, los más pequeños, y la otra mitad al otro, siendo estos más grandes.

Para llevar a cabo el filtrado de mediana en un determinado píxel, lo primero que se hará será ordenar los valores de los píxeles vecinos. Luego determinar la mediana y finalmente asignar este valor al píxel en cuestión.

Cuando varios valores de un entorno o vecindad son iguales, se agruparan como sigue:

Suponiendo un filtro de un entorno en  $3 \times 3$  con valores (10, 20, 20, 15, 20, 25, 20, 100). Estos se ordenan (10, 15, 20, 20, 20, 20, 20, 25, 100) y el valor resultante para el píxel de la imagen final en cuestión será 20.

La siguiente figura representa como queda la imagen después de aplicarle esta técnica de filtrado:



Fig.17: a) Imagen en escala de grises      b) Imagen después del filtro Mediana

Tiene como ventaja que el valor final del píxel es un valor real de la imagen y no un promedio, lo cual reduce el efecto borroso que adquieren las mismas luego de aplicarles un filtro. Este tipo de técnica tiene el inconveniente que es complejo de calcular debido al orden que hay que aplicar a los diferentes valores que aparecen en la imagen para poder determinar el valor central de ellos. Este filtro además de eliminar el ruido también resuelve discontinuidades de esta sin destruir la información general.

### Filtro Media

Este filtro asigna al píxel central la media de todos los píxeles incluidos en la ventana. La matriz de filtrado estaría compuesta por unos y el divisor sería el número total de elementos en la matriz.

El nivel de gris de cada píxel de la imagen final se obtiene mediante el promedio del valor de los niveles de gris de los píxeles de un entorno predefinido. Este entorno es el tamaño del filtro. Por tanto el algoritmo que integra este filtro es el siguiente:

$$g(x,y) = \frac{1}{M} \sum_{(m,n)} f(m,n)$$

En esta ecuación:

- $g(x,y)$  = imagen final.
- $f(m,n)$  = imagen inicial.
- $(m,n)$  = tamaño del filtro.
- $M$  = Suma de los pesos del filtro.

El filtro usado es el siguiente:

$$1/9 \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$$

Este tipo de filtrado provoca que la imagen se vuelva borrosa.

Entre los filtros de media se encuentra el filtro Sigma en el que el valor del píxel es calculado con los valores de un entorno siempre que la diferencia entre el nivel de gris de éstos y la media del entorno no sea superior al doble de la desviación típica de dicho entorno. El algoritmo quedaría de la siguiente forma:

$$g(x,y) = \begin{cases} 1/M \sum_{(m,n)} f(m,n) & \text{si } |f(x,y) - \sum_{(m,n)} f(m,n)| < 2\sigma \\ f(x,y) & \text{de lo contrario} \end{cases}$$

La imagen que se obtiene al aplicar este filtro es la siguiente:



Fig.18: a) Imagen en escala de grises

b) Imagen después del filtro Media

La misión que tienen estos filtros es hacer que aquellos píxeles que resulten afectados de ruido adquieran un valor próximo a los de su entorno, respetando los bordes para no eliminar información.

### Filtro Propuesto

Tras realizar la implementación y prueba de los filtros más utilizados para reducir el ruido de las imágenes es propuesto el filtro gaussiano pues a pesar de que en su procedimiento este tiende a causar el empañamiento de la imagen es uno de los más óptimos, ya que elimina la mayor cantidad de ruido presente en las imágenes.

### 2.2.3 Detección de bordes

Una vez aplicado el filtro a la imagen es necesario detectar los bordes, utilizando para este procedimiento los operadores de contorno. La principal función de estos es realizar el cálculo de un operador local de derivación.

#### Operador Sobel

El detector de contorno Sobel calcula el gradiente de la intensidad de brillo de cada píxel, dando la dirección del mayor incremento posible de negro a blanco. El resultado muestra qué tan abrupto o suave es el cambio producido en la imagen en cada punto analizado, y a su vez que tanto un punto determinado representa un borde en esta y también la orientación a la que tiende ese borde.

Este operador utiliza dos mascarar de 3x3 elementos para aplicar convolución a la imagen original con el objetivo de calcular las aproximaciones a las derivadas. Uno de estos es empleado en los cambios horizontales mientras que el segundo para los verticales.

Si se define A como la imagen original y  $G_x$  y  $G_y$  como las máscaras que representan las derivadas de la intensidad el resultado es calculado como:

$$G_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * A \quad \text{and} \quad G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A$$

En cada punto de la imagen, los resultados de las aproximaciones de los gradientes horizontal y vertical pueden ser combinados para obtener la magnitud del gradiente, mediante:

$$G = \sqrt{G_x^2 + G_y^2}$$

Con esta información es posible calcular también la dirección del gradiente:

$$\Theta = \arctan \left( \frac{G_y}{G_x} \right)$$

Donde  $\Theta$  es 0 para bordes verticales con puntos más oscuros al lado izquierdo.

Este proceso es representado imagen siguiente:



Fig.19: a) Imagen después del filtro Gaussiano    b) Imagen después del operador Sobel

### Operador Prewitt

El operador de Prewitt enfatiza los bordes horizontales y verticales como resultado de aplicar el gradiente para obtener las matrices que constituirán las máscaras. Esta se define por:

$$\Delta x = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad \Delta y = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Estas máscaras hacen convolución con la imagen. En cada píxel ahora se tienen dos medidas:  $s_1$  y  $s_2$ , que corresponden a las máscaras de fila y columna respectivamente, con estas medidas se halla la magnitud y la dirección del borde, que están definidos en siguiente ecuación (Muñoz, Osorio, y Prieto 2001):

$$M = \sqrt{S_1^2 + S_2^2} \quad D = \tan^{-1} \left[ \frac{S_1}{S_2} \right]$$

Con los resultados obtenidos de la aplicación de las máscaras para cada píxel es realizada la misma operación que el operador Sobel: La imagen siguiente muestra el resultado de este procedimiento.



Fig.20: a) Imagen después del filtro Gaussiano b) Imagen después del operador Prewitt

### Operador Canny

Entre los filtros más usados hoy en día se encuentra el operador de Canny, el cual es un operador diferencial combinado a un filtro gaussiano. Este detector de contorno es regido por una serie de pasos explicados a continuación:

1. Reducción de ruido:

La imagen es suavizada empleando un filtro gaussiano mediante el cual son eliminados los detalles y texturas no relevantes en esta.

2. Obtención del gradiente:

Para obtener el gradiente es necesario realizar la aplicación de un filtro gaussiano a la imagen original para suavizarla y eliminar el ruido existente, teniendo precaución en no ser excesivos para no perder detalles de esta y provocar un resultado negativo. El suavizado se obtiene promediando los valores de intensidad de los píxeles en el entorno de vecindad con una máscara de convulsión de media cero y desviación estándar  $\sigma$ . Al finalizar el suavizado se obtiene de cada píxel la magnitud y el módulo del gradiente, obteniendo de este proceso dos imágenes.

(a)	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>1</td><td>4</td><td>7</td><td>4</td><td>1</td></tr> <tr><td>4</td><td>16</td><td>26</td><td>16</td><td>4</td></tr> <tr><td>7</td><td>26</td><td>41</td><td>26</td><td>7</td></tr> <tr><td>4</td><td>16</td><td>26</td><td>16</td><td>4</td></tr> <tr><td>1</td><td>4</td><td>7</td><td>4</td><td>1</td></tr> </table>	1	4	7	4	1	4	16	26	16	4	7	26	41	26	7	4	16	26	16	4	1	4	7	4	1	(b)	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>2</td><td>4</td><td>5</td><td>4</td><td>2</td></tr> <tr><td>4</td><td>9</td><td>12</td><td>9</td><td>4</td></tr> <tr><td>5</td><td>12</td><td>15</td><td>12</td><td>5</td></tr> <tr><td>4</td><td>9</td><td>12</td><td>9</td><td>4</td></tr> <tr><td>2</td><td>4</td><td>5</td><td>4</td><td>2</td></tr> </table>	2	4	5	4	2	4	9	12	9	4	5	12	15	12	5	4	9	12	9	4	2	4	5	4	2
1	4	7	4	1																																																	
4	16	26	16	4																																																	
7	26	41	26	7																																																	
4	16	26	16	4																																																	
1	4	7	4	1																																																	
2	4	5	4	2																																																	
4	9	12	9	4																																																	
5	12	15	12	5																																																	
4	9	12	9	4																																																	
2	4	5	4	2																																																	
$\frac{1}{273}$		$\frac{1}{115}$																																																			

Fig.21: Máscaras de convolución recomendadas para el obtener el filtro gaussiano

### 3. Supresión no máxima:

Las imágenes generadas en el paso anterior sirven de entrada para generar una imagen cuyos bordes sean adelgazados. Se consideran cuatro direcciones:  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  y  $135^\circ$  con respecto al eje horizontal. Es encontrada entonces, para cada píxel, la dirección que mejor se aproxime a la del ángulo de gradiente. A continuación se describe este proceso:

Entrada: imagen I

Máscara de convolución H (con media cero y desviación estándar  $\sigma$ )

Salida: imagen  $E_m$  de la magnitud del gradiente

Imagen  $E_0$  de la orientación del gradiente

- a. Suavizar la imagen I con H mediante un filtro gaussiano y obtener J como imagen de salida.
- b. Para cada píxel (i, j) en J, obtener la magnitud y orientación del gradiente basándose en las siguientes expresiones: El gradiente de una imagen  $f(x,y)$  en un punto (x,y) se define como un vector bidimensional dado por la ecuación:

$$G[f(x, y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} f(x, y) \\ \frac{\partial}{\partial y} f(x, y) \end{bmatrix}$$

Siendo un vector perpendicular al borde, donde el vector G apunta en la dirección de variación máxima de f en el punto (x,y) por unidad de distancia, con la magnitud y dirección dadas por:

$$|G| = \sqrt{G_x^2 + G_y^2} = |G_x| + |G_y| ,$$

$$\phi(x, y) = \tan^{-1} \frac{G_y}{G_x}$$

- c. Obtener  $E_m$  a partir de la magnitud de gradiente y  $E_0$  a partir de la orientación según las expresiones anteriores.

Posteriormente se observa si el valor de la magnitud de gradiente es más pequeño que al menos uno de sus dos vecinos en la dirección del ángulo obtenida en el paso anterior. De ser así se asigna el valor 0 a dicho píxel, en caso contrario se asigna el valor que tenga la magnitud del gradiente.

### 4. Histéresis de umbral a la supresión no máxima:

Con el paso anterior la imagen obtenida puede contener máximos locales creados por el ruido. Esto puede ser solucionado con las histéresis del umbral, cuyo proceso consiste en dada la imagen concebida, tomar la orientación de los puntos de borde de la misma y dos umbrales, siendo el primero menor que el segundo. Para cada punto se debe localizar el siguiente punto de borde no explorado que sea mayor al segundo umbral. A partir de dicho punto seguir las cadenas de máximos locales conectados en ambas direcciones perpendiculares a la normal del borde, siempre que estas sean mayores al primer umbral. De esta manera se marcan los puntos visitados y se agregan a una lista de los puntos de contorno conectados. Y así se eliminan las uniones en forma de Y de los segmentos que confluyan en un punto. El algoritmo de supresión no máxima se explica a continuación.

Entrada: imagen  $m \times E$  de la magnitud del gradiente.

Imagen  $o \times E$  de la orientación del gradiente.

Salida: imagen  $n \times I$

Considerar: cuatro direcciones  $d_1, d_2, d_3, d_4$  identificadas por las direcciones de  $0^\circ, 45^\circ, 90^\circ$  y  $135^\circ$  con respecto al eje horizontal.

a) Para cada píxel  $(i, j)$ :

- Encontrar la dirección  $d_k$  que mejor se aproxima a la dirección  $E_0(i, j)$ , que viene a ser la perpendicular al borde.
- Si  $E_m(i, j)$  es más pequeño que al menos uno de sus dos vecinos en la dirección  $d_k$ , al píxel  $(i, j)$  de  $I_n$  se le asigna el valor 0,  $I_n(i, j) = 0$  (supresión), de otro modo  $I_n(i, j) = E_m(i, j)$
- Devolver  $I_n$

### 5. Cierre de contornos abiertos:

Consiste en buscar los extremos de los contornos abiertos siguiendo la dirección del máximo gradiente hasta cerrarlos con otro extremo abierto.

Entrada: imagen  $I_n$  obtenida del paso anterior.

Imagen  $E_0$  de la orientación del gradiente umbral  $t_1$  umbral  $t_2$ , donde  $t_1 < t_2$

Salida: imagen  $G$  con los bordes conectados de contornos

a) Para todos los puntos de  $I_n$  y explorando  $I_n$  en orden fijo:

- i. Localizar el siguiente punto de borde no explorado previamente,  $I_n(i, j)$  tal que  $I_n(i, j) > t_2$
  - ii. Comenzar a partir de  $I_n(i, j)$ , seguir las cadenas de máximos locales conectados en ambas direcciones perpendiculares a la normal de borde, siempre que  $I_n > t_1$ .
  - iii. Marcar todos los puntos explorados y, salvar la lista de todos los puntos en el entorno conectado encontrado.
- b) Devolver  $G$  formada por el conjunto de bordes conectados de contornos de la imagen, así como la magnitud y orientación, describiendo las propiedades de los puntos de borde.

Al finalizar este proceso se obtiene como resultado una imagen binaria en la que cada píxel se define como píxel de borde o como píxel no perteneciente al borde. Las siguientes figuras son la muestra de realizar el operador de Canny.



Fig.22: a) Imagen después del filtro Gaussiano b) Imagen después del operador Canny

### Operador Roberts

El algoritmo de Roberts parte de la comparación cruzada de cuatro píxeles pero realizando la comparación como el resultado de una suma vectorial.

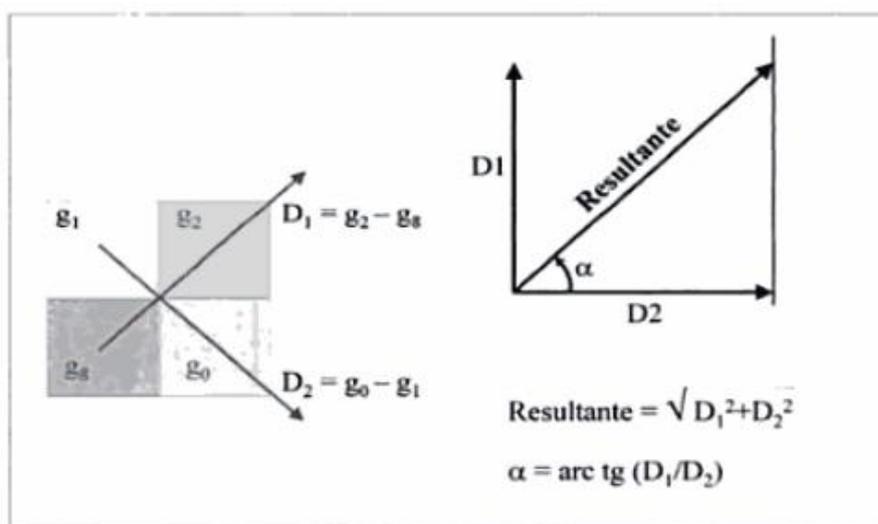
Es uno de los más antiguos, simples y menos utilizado de los métodos de detección de bordes en la actualidad, a pesar de que su sencillez en la implementación es atrayente para las aplicaciones hardware de alta velocidad. Este operador utiliza dos máscaras de convolución.

## CAPÍTULO 2: ALGORITMOS PROPUESTOS

$$F_x = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \quad F_y = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Tomando la diferencia cruzada D1 y D2 como vectores, el módulo de cada uno de ellos se puede representar como la magnitud, mientras que el signo se puede interpretar como la dirección.

Los vectores resultantes obtenidos como diferencias entre los píxeles cruzados forman un ángulo de  $45^\circ$  con la horizontal. Sus magnitudes, representadas como D1 y D2 forman un ángulo de  $90^\circ$  y serán positivos o negativos según los minuendos sean mayores o menores que los sustraendos. Teniendo en cuenta la magnitud del signo, se podrá obtener la suma de ambos, que será la resultante. Si se aplica Pitágoras, la magnitud de la resultante será el valor de la hipotenusa del triángulo que tiene como catetos D1 y D2, con una dirección igual al arco cuya tangente es D1/D2, que indica precisamente la dirección del margen (Grau, José F. Pertusa 2003).



La ventaja de este operador es que es fácil y rápido de computar. Sólo está implicado un entorno de vecindad de 4 píxeles y sólo se usan sumas y restas en los cálculos y su principal desventaja es que, si lo que se quiere es determinar bordes, es muy sensible al ruido y tiene una respuesta débil a los verdaderos bordes, a menos que sean muy pronunciados. Para este propósito funciona mejor el operador de Sobel.



Fig.23: a) Imagen después del filtro Gaussiano b) Imagen después del operador Robert

### Operador ImageFilter

La biblioteca PIL es un complemento disponible a utilizar en el intérprete de Python. Esta permite diversos procedimientos con las imágenes tal como la conversión de imágenes de diferentes píxeles. Para sus procedimientos utiliza la función `convert ('L')`. Esta librería soporta transformaciones en los modos de las imágenes.

La biblioteca Imaging de Python agrega capacidades de imágenes al intérprete de Python. Esta proporciona apoyo de formato de archivo extenso, una representación interna eficaz y un procesamiento poderoso de imágenes. Contiene funcionalidades básicas de las imágenes tales como filtrado, convolución y conversión espacial de colores

El modulo ImageFilter perteneciente a esta librería contiene definiciones para un conjunto predefinido de filtros:

- `Kernel`: Crea una máscara de convolución del tamaño dado.
- `RankFilter`: Crea un rango de filtros del tamaño especificado.
- `MinFilter`, `MedianFilter`, `MaxFilter`: Crea un filtro según el tamaño especificado, ya sea mínimo, mediano o máximo.
- `ModeFilter`: Crea un modo de filtro del tamaño otorgado.

A continuación se muestra la imagen resultante del filtro definido por el módulo ImageFilter.



Fig.24: a) Imagen después del filtro Gaussiano b) Imagen después del operador ImageFilter

### Detectores de contorno propuestos

Se escogieron los detectores de contorno de Sobel y Prewitt ya que estos utilizan máscaras de convolución de 3x3 que les permiten detectar mayor cantidad de bordes a diferencia de los otros métodos investigados. Además estos algoritmos utilizan la segunda derivada la cual permite suavizar la imagen de manera gradual.

### **Conclusiones**

En este capítulo se trazaron los pasos a seguir para el desarrollo de los algoritmos para la detección y localización de texto en imágenes. Además se realizó una descripción detallada del funcionamiento de los mismos. Finalmente se seleccionaron los algoritmos a utilizar para dar solución a la problemática planteada. Fueron elegidos los detectores de contornos de Sobel y Prewitt, estas técnicas se consideran satisfactorias en cuanto a velocidad de procesamiento, factibilidad de implementación, tiempo y recursos.

## **Introducción**

En este capítulo se detallan los componentes que integran la clase que permite realizar la detección y localización de texto en imágenes, además se describe la conexión existente entre ellos. Es explicada la integración de los algoritmos implementados con el producto Ecúmene Pyxel. Además son realizadas las pruebas necesarias que permitan comprobar la validez de los algoritmos de detección implementados.

### **3.1 Implementación**

En este flujo de trabajo se convierten las clases y objetos en ficheros fuente, binarios, ejecutables entre otros. La finalidad principal es alcanzar la capacidad operacional del producto de forma incremental a través de las sucesivas iteraciones. Además todos los componentes, características y requisitos deben ser implementados, integrados y probados en su totalidad, obteniendo una versión aceptable del producto.

### **3.1 Diagrama de Componentes**

El diagrama de componentes ilustra los elementos que serán usados para construir el sistema. UML define cinco estereotipos estándar que se le aplican a los componentes: ejecutable, bibliotecas, tabla, archivo y documento. Los distintos componentes se pueden agrupar en paquetes según un criterio lógico y con vista a simplificar su implementación. Estos trabajan acoplados, brindando el comportamiento que se requiere para el cumplimiento de los requerimientos expuestos en capítulos anteriores.

Desde el punto de vista de este diagrama se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización, y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo. Los elementos de modelado dentro de un diagrama de componentes serán componentes y paquetes. En la siguiente figura se muestra el diagrama de componentes y las relaciones que existen entre ellos.

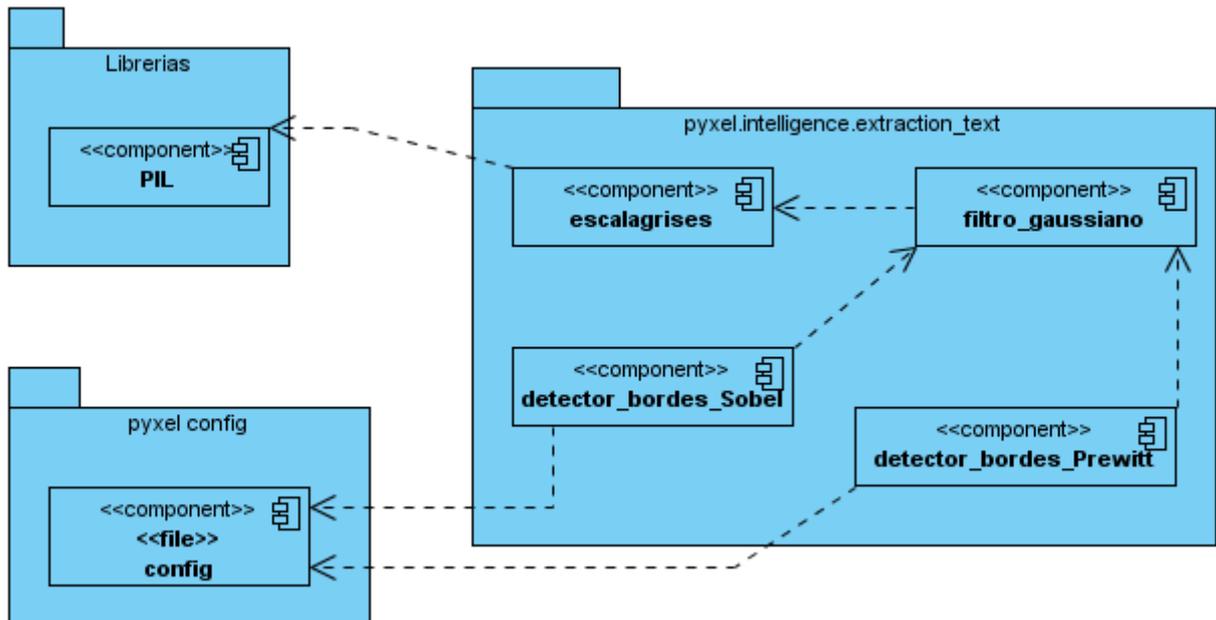


Fig.25: Diagrama de componentes

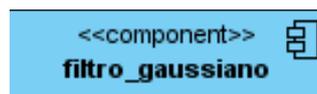
Para una mejor comprensión de todos los componentes representados en la Fig. 26 se es necesario conocer qué función realiza cada uno de ellos.

En la carpeta `pyxel.intelligence.extraction_text` se encuentra los componentes `escalagrises.py`, `filtro_gaussiano.py`, `detector_bordes_Sobel`, `detector_bordes_Prewitt`. A continuación se explica cada uno de ellos detalladamente:

- **escalagrises.py**: módulo encargado de convertir las imágenes de color a escala de grises auxiliándose de la librería PIL.



- **filtro gaussiano**: módulo a cargo de reducir el ruido en las imágenes.



- **detector\_bordes\_Sobel**: módulo encargado de detectar los bordes para lograr resaltar las regiones que contiene texto en las imágenes.



- **detector\_bordes\_Prewitt:** otro módulo encargado de detectar los bordes y de esta manera resaltar las regiones con texto en las imágenes.



Estos componentes se encuentran estrechamente relacionados con la librería PIL para el desarrollo de la aplicación y con el sistema en general.

- **PIL(Python Imaging Library):** librería que permite manipular imágenes de manera eficiente.



- **Config:** Carpeta perteneciente al sistema Ecúmene Pyxel encargada de manejar los servicios existentes.



### 3.2 Integración con Ecúmene Pyxel

Ecúmene Pyxel es un sistema distribuido compuesto por varios subsistemas que pueden correr en varias máquinas y comunicarse a través de la red. Esta distribución aporta varias ventajas:

- El sistema es flexible y tolerante a los fallos que puedan presentarse en uno de los subsistemas sin que esto dañe al sistema en general. Esto también permite que los errores puedan ser corregidos con mayor facilidad.
- No es necesario que todos los módulos estén instalados en todas las máquinas, sino sólo los módulos que componen el o los subsistemas que estén instalados en cada máquina.
- Es posible modificar o reemplazar un subsistema sin la necesidad de detener el sistema completo

El subsistema *pyxel.intelligence.extraction\_text* es uno de los diversos subsistemas que compondrán Pyxel.

Pyxel depende de Pyro para la comunicación entre los diferentes subsistemas existentes, además cuenta con una capa de abstracción sobre Pyro, llamada pyxel.launcher, que simplifica el uso de Pyro y logra desacoplar la implementación de los servicios y nodos de Pyro.

### 3.2 Pruebas realizadas

Las pruebas realizadas están enfocadas en demostrar que los detectores de contorno Sobel y Prewitt son buenos manipuladores de imágenes, de manera especial para la detección de texto. Por diversos motivos explicados a continuación se hace necesario asegurar que Prewitt es un detector más adecuado y eficiente para la detección de bordes que Sobel. La implementación de ambos algoritmos fue realizada en Python utilizando como plataforma UNIX y como estación de trabajo Debian Linux, Intel(R) Core(TM)2 Duo CPU 2.00Hz.

Para la realización de dichos experimentos se efectuó la selección de una colección de imágenes mostradas a continuación.



Fig.26 Colección de imágenes empleada. Enumeradas de 1-6 de Izq. a Der. de arriba hacia abajo

### PRUEBA 1

En el presente experimento se requiere demostrar cuál de los detectores propuestos por esta investigación es más eficiente basándose en el tiempo de demora de la detección total de bordes en una imagen. Esto es conseguido mediante el cálculo del tiempo en que demora una función en ejecutarse mediante el módulo time ofrecido por la librería PIL de Python. En esta se representan las siguientes variables:

Ti: Tiempo Inicial

TF: Tiempo Final

TT = TF – Ti, Tiempo total

Detector Sobel	T <sub>i</sub>	T <sub>f</sub>	T <sub>T</sub>
Imagen 1	0.03	10.85	10.82
Imagen 2	0.03	23.34	23.31
Imagen 3	0.03	10.53	10.50
Imagen 4	0.02	10.85	10.82
Imagen 5	0.03	29.24	29.21
Imagen 6	0.04	26.53	26.49

**Tabla 1: Tiempo de ejecución del detector Sobel**

Detector Prewitt	T <sub>i</sub>	T <sub>f</sub>	T <sub>T</sub>
Imagen 1	0.03	11.25	11.19
Imagen 2	0.03	25.28	25.25
Imagen 3	0.04	10.58	10.55
Imagen 4	0.02	11.25	11.19
Imagen 5	0.03	34.80	34.78
Imagen 6	0.03	26.84	26.81

**Tabla 2: Tiempo de ejecución del detector Prewitt**

Al finalizar el experimento realizado se concluye que el operador Sobel es más eficiente en cuanto al tiempo de ejecución que el operador Prewitt debido a los resultados de los experimentos realizados a la colección de imágenes, mostrado en las tablas 1 y 2.

### PRUEBA 2

La segunda prueba realizada está basada en la detección de bordes de la imagen. En esta se realizará un conteo de los bordes detectados por cada uno de los detectores de contorno demostrando cuál de ellos es más óptimo.

En este experimento se tendrán presente las siguientes variables:

$C_x$ : cantidad de bordes detectados en el eje vertical  $x$ .

$C_y$ : cantidad de bordes detectados en el eje vertical  $y$ .

$C_B$ : cantidad de bordes detectados.

Detector Sobel	$C_x$	$C_y$	$C_B$
Imagen 1	117512169	346040731	463552900
Imagen 2	605679409	1756443765	2362123174
Imagen 3	65649989	172817857	235467846
Imagen 4	307324693	896839129	1204163822
Imagen 5	188700202	533676842	722377044
Imagen 6	337471854	984984454	1322456308

**Tabla 3: Cantidad de bordes detectados del detector Sobel**

Detector Prewitt	$C_x$	$C_y$	$C_B$
Imagen 1	89784227	261206905	350991132
Imagen 2	465262999	133133439	1796597348
Imagen 3	50224889	130766463	180991322
Imagen 4	236238161	679483247	915721408
Imagen 5	148401756	405308176	553709932
Imagen 6	257955324	743515032	1001470356

**Tabla 4: Cantidad de bordes detectados del detector Prewitt**

Los resultados de los experimentos realizados muestran que el detector de contorno Sobel es más óptimo en la detección de bordes que el detector Prewitt. Estos resultados son mostrados en las tablas 3 y 4.

### **Conclusiones**

En este capítulo se evaluaron los algoritmos propuestos mediante la aplicación de dos pruebas, la primera de ellas en cuanto al tiempo de ejecución y la segunda en cuanto a la cantidad de bordes detectados, lo cual permitió analizar la eficiencia de cada uno de ellos.

Con el fin del presente capítulo se da por terminada la propuesta de solución relacionada al desarrollo de los algoritmos inherentes al presente trabajo de diploma: “Implementación de algoritmos para la detección y localización de texto en imágenes”.

La gestión de imágenes y en específico la búsqueda eficiente de estas es un elemento fundamental en los medios de prensa debido a la importancia que las imágenes tienen en las noticias y la necesidad de su inmediatez. La propuesta de solución introduce mejoras en las búsquedas de imágenes, ya que permitirá agilizar la búsqueda AL consultar la información contenida en las ellas.

Por lo tanto se puede concluir que con el desarrollo de este trabajo se le da cumplimiento a los objetivos planteados:

- Se valoraron cinco algoritmos para la detección y localización de texto en imágenes de los cuales se seleccionaron los dos más eficientes.
- Se eligieron los lenguajes, herramientas y metodología a emplear en la elaboración de la propuesta de solución, los cuales están acorde a la política de informatización de la Universidad de las Ciencias Informáticas y con las necesidades de los medios de prensa.
- Se implementaron dos algoritmos de detección y localización de texto en imágenes que tras realizar un procedimiento de mejora a la imagen realza los bordes de la misma, lo cual facilitará en gran medida la búsqueda de imágenes en los medios de prensa cubanos.
- Se evaluaron los algoritmos en cuanto a velocidad de procesamiento y cantidad de bordes detectados.
- Se integraron los algoritmos implementados al producto Pyxel para facilitar la gestión de imágenes en los medios de prensa en Cuba.

Para optimizar las búsquedas de imágenes es necesario utilizar al máximo los metadatos que están contenidos en esta. Para completar el proceso de acelerar las búsquedas de imágenes es necesario aplicar a la imagen un software Reconocedor Óptico de Caracteres (OCR) que permita extraer el texto insertado en las mismas. Por lo tanto se recomienda:

- La continuación de la presente investigación con la implementación de métodos que permitan la extracción del texto anteriormente detectado.
- La integración de los mismos con el producto Ecúmene Pyxel para posteriormente utilizarlo en los medios de prensa cubanos.

**CIPRE:** Centro de Información para la Prensa.

**Fujos de Trabajo:** es el estudio de los aspectos operacionales de una actividad de trabajo: cómo se estructuran las actividades, cómo se realizan, cuál es su orden correlativo, cómo se sincronizan, como fluye la información que soporta las tareas y como se le realiza seguimiento al cumplimiento de estas.

**GNU GPL:** Licencia Pública General (GPL: General Public License) es una licencia fundada en 1989 para proteger la libre distribución, modificación y uso del Software Libre, creada por la Free Software Foundation.

**Java:** Lenguaje de programación orientado a objetos con el que se puede realizar cualquier tipo de programa, es un lenguaje muy extendido, es un lenguaje independiente de la plataforma, es compilado en un bytecode que es interpretado desarrollado por la compañía Sun Microsystems a principios de los 90.

**Metodología:** Es un conjunto de métodos de investigación apropiados al quehacer de una ciencia determinada. Es la parte de un proceso de investigación que permite sistematizar los métodos y técnicas para realizar el proceso con eficiencia.

**Multiparadigma:** Término usado para referirse a los programas, sistemas operativos, lenguajes de programación, u otra clase de software, que puedan funcionar en diversas plataformas.

**OCR:** El Reconocedor Óptico de Caracteres identifica automáticamente caracteres o símbolos que pertenecen a un determinado alfabeto, a partir de una imagen para almacenarla en forma de datos con los que podremos interactuar mediante un programa de edición de texto o similar.

**Producto:** es cualquier cosa que puede ser ofrecida al mercado para su compra, para su utilización o para su consideración. Es cualquier bien, servicio o idea capaz de motivar y satisfacer a un comprador.

**Plataformas:** Es una combinación de hardware y software usada para ejecutar aplicaciones; en su forma más simple consiste únicamente de un sistema operativo, una arquitectura, o una combinación

de ambos. Programación Orientada a Objetos (POO): Es un paradigma de programación que usa objetos y sus interacciones para diseñar aplicaciones y programas de computadora.

**Programación:** Proceso de creación de un programa de computadora, mediante la aplicación de procedimientos lógicos.

**Servicio:** Los servicios son simplemente procesos que están corriendo permanentemente.

**Software:** Conjunto de programas elaborados por el hombre, que controlan la actuación del computador, haciendo que éste siga en sus acciones una serie de esquemas lógicos predeterminados y pueda desempeñar tareas inteligentes. Suele sustituirse por expresiones tales como programas (informáticos) o aplicaciones (informáticas).

**UPEC:** Unión de Periodistas de Cuba.

**UCI:** Universidad de las Ciencias Informáticas.

Grau, José F. Pertusa. Técnicas de análisis de imagen. 2003.

Gubern, Román. Del bisonte a la realidad virtual. 1996. [citado el 21 Abril del 2010]. Disponible en: <http://www.scribd.com/doc/4085734/Del-bisonte-a-la-realidad-virtual>.

Informáticas, Universidad de las Ciencias. Especificación funcional Pyxel 1.0. 2008.

Jacobson, I., y G. Booch. *El proceso unificado de desarrollo de software*. [La Habana]: Félix Varela, 2004.

Lugo, P. Detección de Zonas de Diferentes Texturas Usando Análisis de Multiresolución.

Muñoz, Alejandro, Gustavo Osorio, y Flavio Prieto. Detección de patologías celulares empleando visión artificial. 2001.

Rebaza, Jorge Valverde. Detección de bordes mediante el algoritmo de Canny. Disponible en: <http://www.seccperu.org/files/Detecci%C3%B3ndeBordes-Canny.pdf>.

Rodríguez, Hugo. Curso de Imagen Digital -. 2003. [citado 5 May 2010]. Disponible en: [http://www.hugorodriguez.com/cursos/curso-idigital\\_01.htm](http://www.hugorodriguez.com/cursos/curso-idigital_01.htm).

1. ProcesamientoImágenesDigitales.pdf (Objeto application/pdf). [21 Abril 2010]. Disponible en:<http://www.itpuebla.edu.mx/Eventos/MemoriasyResSemanaInformativa2007/29-procesamientoImágenesDigitales.pdf>
2. Sobre la Historia y la Estética de la Imagen Digital.pdf. [21 Abril 2010]. Disponible en:<http://www.pucsp.br/~gb/texts/SOBRE%20LA%20HISTORIA%20Y%20LA%20ESTETICA%20ODE%20LA%20IMAGEN%20DIGITAL.pdf>
3. Clark, P., y M. Mirmehdi. Location y Recovery of Text on Oriented Surfaces.
4. Coto, Ernesto. Métodos de Segmentación de Imágenes Médicas. Marzo 2003.
5. Chen, Datong, Jean-Marc Odobez, y Herv/e Bourlard. Text detection y recognition in images y video frames. Junio 2003.
6. Gadea, Moisés Pastor. Aportaciones al Reconocimiento Automático de Texto Manuscrito. de abril de 2007.
7. García, Cristián Arriagada, y Diego Aracena Pizarro. Detección y reconocimiento de señales de tránsito utilizando matching de chamfer. Junio 2007.
8. González, Ailyn Alfonso, y Joel Armada Herrera. Creación de un sitio modelo para el desarrollo de portales web utilizando el CMS Plone. 2008.
9. Grupo de tecnología industrial. Detección de bordes en una imagen.
10. Gu, Lifang. Text Detection y Extraction in MPEG Video Sequences. Septiembre 2001.
11. Hermosilla, T., E. Bermejo, A. Balaguer, y L.A. Ruiz. Detección de bordes con precisión subpíxel en imágenes digitales: Interpolación lineal frente a esquemas de tipo no lineal. Noviembre 2006.
12. Hua, Xian-Sheng. Automatic Location of Text in Video Frames.
13. Hua, Xian-Sheng, Pei Yin, y Hong-Jiang Zhang. Efficient video text recognition using multiple frame integration.
14. Informáticas, Universidad de las Ciencias. Especificación funcional Pyxel 1.0. 2008.
15. Jung, Keechul, Kwang In Kim, y Anil K. Jain. Text Information Extraction in Images y Video: A Survey.
16. Lienhart, Rainer, y Wolfgang Effelsberg. Automatic Text Segmentation y Text Recognition for Video Indexing. Mayo 2008.
17. López, Alfonso, y Francisco Sánchez. Bancos de imágenes: fototecas en medios de comunicación.

18. Lugo, P. Detección de Zonas de Diferentes Texturas. Uso Análisis de Multiresolución.
19. Mazzaferri, J. Método óptico para la detección omnidireccional de bordes.
20. Méndez, Alejandro Flores, y Ana Antígona Méndez Cuanalo. Detección estable de los bordes de la oreja en imágenes 2D.
21. Morán, M. A. Jaramillo, J. A. Fernández Muñoz, y E. Martínez de Salazar Martínez. Detección de discontinuidades y mejora de contrastes mediante el cálculo de gradientes uso redes neuronales celulares.
22. Muñoz, Alejandro Flores, Gustavo Osorio, Favio Prieto, y Fabiola Angulo. Detección de patologías celulares empleo visión artificial. Septiembre 2005.
23. Quintas, Manuel Mazo. Visión por computador.
24. Quiroz, José Moreira. Implementación de un algoritmo para la detección y conteo de células en imágenes microscópicas.
25. Rebaza, Jorge Valverde. Detección de bordes mediante el algoritmo de Canny. Disponible en: <<http://www.seccperu.org/files/Detecci%C3%B3ndeBordes-Canny.pdf>>.
26. Suarez, Yoel Rivera, y Tourt, Fernando Echemendía. Sistema para la Detección y Extracción de Textos en Videos Digitales. Mayo 2008.
27. Tekinalp, Serhat. Detecting y recognizing text from video frames. Septiembre 2005.
28. Wolf, Christian, y Jean-Michel Jolion. Model based text detection in images y videos: a learning approach.
29. Filtros para detectar bordes. [21 Abril 2010]. Disponible en: <file:///D:/Usuarios/Ana/Tesis/Bibliograf%C3%ADa/navegadores/filters-edge.html>.
30. Black-byte » Detección de Bordes. [21 Abril 2010]. Disponible en: <file:///D:/Usuarios/Ana/Tesis/Bibliograf%C3%ADa/navegadores/Black-byte%20%C2%BB%20Deteccion%20de%20Bordes.htm>.
31. Definición de Píxel y concepto » Definición ABC. [21 Abril 2010]. Disponible en: <<http://www.definicionabc.com/tecnologia/pixel.php>>.
32. Detección de Bordes. [21 Abril 2010]. Disponible en: <file:///D:/Usuarios/Ana/Tesis/Bibliograf%C3%ADa/navegadores/bordes.html>.
33. Disponible Geany 0.18.1, un ligero y sencillo IDE. [21 Abril 2010]. Disponible en: <<http://www.noticiasubuntu.com/disponible-geany-0-18-1-un-ligero-y-sencillo-ide/>>.
34. La detección de discontinuidades. [21 Abril 2010]. Disponible en: <file:///D:/Usuarios/Ana/Tesis/Bibliograf%C3%ADa/navegadores/detectdisc.aspx.htm>.

35. Object-tracking con Textura en Imágenes: C Wavelet. [21 Abril 2010]. Disponible en: <http://texturacomplexwaveletfi.blogspot.com/>.
36. Oscar Fernández Asunción: Detección de Bordos: algoritmo de Canny. [21 Abril 2010]. Disponible en: <file:///D:/Usuarios/Ana/Tesis/Bibliograf%C3%ADa/navegadores/deteccion-de-bordes-algoritmo-de-canny.html>.
37. Práctica Detección de Bordos. [21 Abril 2010]. Disponible en: [file:///D:/Usuarios/Ana/Tesis/Bibliograf%C3%ADa/navegadores/P\\_Bordes.html](file:///D:/Usuarios/Ana/Tesis/Bibliograf%C3%ADa/navegadores/P_Bordes.html).
38. Procesamiento digital de imágenes - Wikipedia, la enciclopedia libre. [21 Abril 2010]. Disponible en: [http://es.wikipedia.org/wiki/Procesamiento\\_digital\\_de\\_im%C3%A1genes#imagen](http://es.wikipedia.org/wiki/Procesamiento_digital_de_im%C3%A1genes#imagen).
39. Recuperación y organización de la información - Recuperación de imágenes y vídeos. [21 Abril 2010]. Disponible en: [http://recuperaciondeimagenesyvideos.50webs.com/recuperacion\\_de\\_imagenes.html](http://recuperaciondeimagenesyvideos.50webs.com/recuperacion_de_imagenes.html).
40. The Image Filter Module. [21 Abril 2010]. Disponible en: <file:///D:/Usuarios/Ana/Tesis/Bibliograf%C3%ADa/navegadores/imagefilter.htm>.
41. Gubern, Román. Del bisonte a la realidad virtual. 1996. [21 Abril 2010]. Disponible en: <http://www.scribd.com/doc/4085734/Del-bisonte-a-la-realidad-virtual>.
42. Kruchten, P. The rational unified process: an ... - Google Libros. 2003. [21 Abril 2010]. Disponible en: [http://books.google.com.cu/books?id=RYCvMx6o47pMC&dq=The+Rational+Unified+Process+An+Introduction&printsec=frontcover&source=bn&hl=es&ei=uMPPS4Tdl4GK8wSinJWxOw&sa=X&oi=book\\_result&ct=result&resnum=4&ved=0CCAQ6AEwAw#v=onepage&q&f=false](http://books.google.com.cu/books?id=RYCvMx6o47pMC&dq=The+Rational+Unified+Process+An+Introduction&printsec=frontcover&source=bn&hl=es&ei=uMPPS4Tdl4GK8wSinJWxOw&sa=X&oi=book_result&ct=result&resnum=4&ved=0CCAQ6AEwAw#v=onepage&q&f=false).
43. Teorema fundamental del cálculo para el gradiente. [26 Abril 2010]. Disponible en: <http://www.megascience.com/fisica/ElectricidadMagnetismo/TeoremaFundamentalGradiente/Default.aspx>.
44. Grau, José F. Pertusa. Técnicas de análisis de imagen. 2003.
45. Muñoz, Alejandro, Gustavo Osorio, y Flavio Prieto. Detección de patologías celulares empleando visión artificial. 2001.

### ANEXO 1: Cálculo del Gradiente

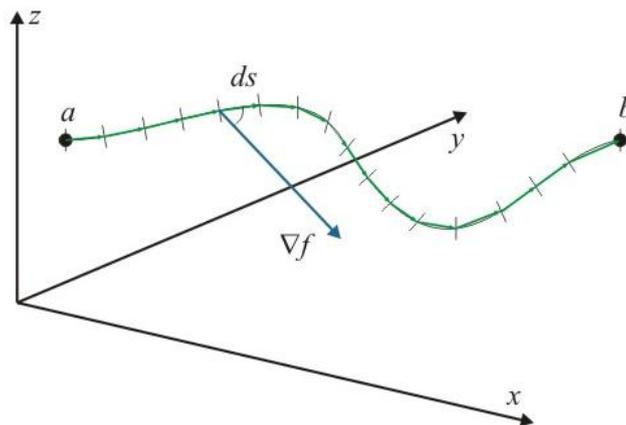
Sea  $f(x, y, z)$  una función escalar (o campo escalar) y sean  $a(x_a, y_a, z_a)$  y  $b(x_b, y_b, z_b)$  dos puntos del espacio. Si se considera una curva  $C$  cualquiera que vaya de  $a$  a  $b$ .

La integral de línea a lo largo de la curva  $C$  es igual al incremento de la función  $f$  de  $a$  a  $b$ , esto es:

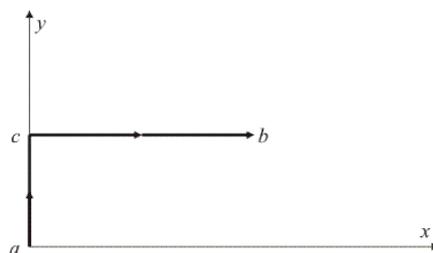
$$\int_a^b (\nabla f) \cdot ds = f(b) - f(a)$$

Donde  $f(a) = f(x_a, y_a, z_a)$  y  $f(b) = f(x_b, y_b, z_b)$

El producto escalar  $(\nabla f) \cdot ds$  es el incremento infinitesimal de  $f$ . Al hacer la integral se suman los distintos incrementos de  $f$  por lo que la suma total nos dará el incremento de  $f$  de  $a$  a  $b$ , es decir  $f(b) - f(a)$ .



Como ejemplo se chequea el teorema fundamental del cálculo para el gradiente utilizando la función  $f(x, y) = xy^2$  y los puntos  $a(0,0,0)$   $b(2,1,0)$  a lo largo de la curva  $acb$  formada por segmentos como se indica en la figura, siendo  $c(0,1,0)$ .



Por un lado se tiene, para el segundo miembro del teorema fundamental  $f(b) - f(a) = 2(1)^2 - 0(0)^2 = 2$

A continuación se calcula el primer miembro del teorema fundamental. El primer paso es determinar el gradiente de  $f$ :  $\nabla f = y^2 \mathbf{x} + 2xy \mathbf{y}$

Recordando que  $d\mathbf{s} = dx \mathbf{x} + dy \mathbf{y} + dz \mathbf{z}$  nos queda  $(\nabla f) \cdot d\mathbf{s} = y^2 dx + 2xy dy$  entonces:

$$\int_a^b (\nabla f) \cdot d\mathbf{s} = \int_a^b (y^2 dx + 2xy dy)$$

Para calcular esta integral se divide la curva en dos, una formada por el segmento vertical  $ac$  y al otro por el segmento horizontal  $cb$ :

$$\int_a^b (y^2 dx + 2xy dy) = \int_a^c (y^2 dx + 2xy dy) + \int_c^b (y^2 dx + 2xy dy)$$

Se calcula primero la integral:

$$\int_a^c (y^2 dx + 2xy dy)$$

Para los puntos del segmento vertical  $ac$  no incremento en  $x$  por lo que  $dx = 0$ . Además todos los puntos del segmento tienen igual valor de  $x$  por lo que  $x = cte. = 0$  así que:

$$\int_a^c 2xy dy = \int_{y_a}^{y_c} 2xy dy = 2x \int_0^1 y dy = 2(0) \int_0^1 y dy = 0$$

El siguiente paso sería:

$$\int_c^b (y^2 dx + 2xy dy)$$

Para los puntos del segmento horizontal  $cb$  no incremento en  $y$  por lo que  $dy = 0$ . Además todos los puntos del segmento tienen igual valor de  $x$  por lo que  $x = cte. = 1$  así que:

$$\int_c^b (y^2 dx + 2xy dy) = \int_c^b y^2 dx = y^2 \int_{x_c}^{x_b} dx = 1^2 \int_0^2 dx = 2$$

Finalmente se obtiene:

$$\int_a^b (\nabla f) \cdot d\mathbf{s} = 0 + 2 = 2$$