

# Algoritmos y Estructuras de Datos III

Departamento de Computación  
Facultad de Ciencias Exactas y Naturales  
Universidad de Buenos Aires

## Trabajo Practico 3

### Grupo 15

Integrante	LU	Correo electrónico
Mariano Javier Cerrutti	525/07	vscorza@gmail.com
Federico xxx Huel	xxx/07	federico.huel@gmail.com
Rogelio xxx Mita	xxx/07	rogeliomita@gmail.com

### Reservado para la catedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

# Índice

<b>1. Presentación</b>	<b>3</b>
<b>2. Propuesta</b>	<b>4</b>
<b>3. Algoritmo</b>	<b>5</b>
3.1. Introducción . . . . .	5
3.2. Desarrollo . . . . .	5
3.3. Pseudocódigo . . . . .	5
3.4. Detalles de implementación . . . . .	5
3.5. Cálculo de complejidad . . . . .	6
3.6. Demostración . . . . .	7
3.7. Análisis . . . . .	7
<b>4. Conclusiones</b>	<b>8</b>

## 1. Presentación

El propósito de este trabajo es presentar una solución al problema de *detección de bordes en una imagen digital* a través de un operador definido por una matriz de convolución.

Acá hay un poco de sarasa matemática:

Sea  $p$  pixel,  $|p| = 4$  y una imagen  $I = p_{i,j} : i \in ancho, j \in alto$  se aplicará el operador sobel en  $x$

$$\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$$

## 2. Propuesta

Acá mandamos una propuesta? No se como armamos en general el reporte.

Un grafo por si hay que meter alguno después

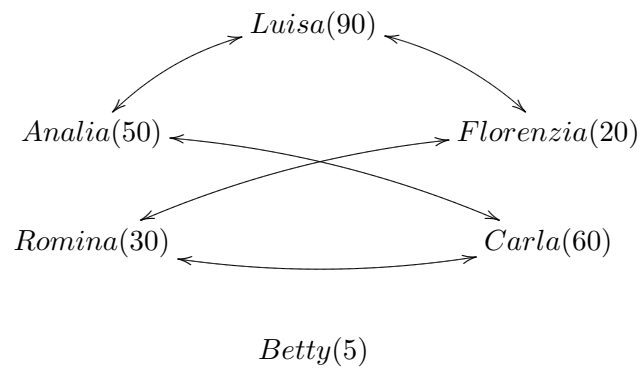


Figura 2.1: El grafo de intimidades.

### 3. Algoritmo

#### 3.1. Introducción

Acá metemos la descripción del algoritmo.

#### 3.2. Desarrollo

Explicamos el algo.

#### 3.3. Pseudocódigo

Acá metemos el pseudo.

Esto será recorrido así, sarasa de pseudocódigo:

---

##### Algoritmo 1 SobelEnX

---

```

1:  $I \leftarrow p_{i,j} : i \in ancho, j \in alto$ 
2: para  $i \in ancho$  hacer
3:   para  $j \in alto$  hacer
4:      $p_{i,j} \leftarrow -p_{i-1,j-1} - 2 * p_{i-1,j} - p_{i-1,j+1} + p_{i+1,j-1} + 2 * p_{i+1,j} + p_{i+1,j+1}$ 
5:   fin para
6: fin para

```

---

#### 3.4. Detalles de implementación

Acá puede ir todo o parte del asm.

```

1  asmSobel:
2      doEnter 1
3
4      cmp dword XORDER, 0
5      je     ySobel
6
7      xSobel:
8      ;=====
9      ; SOBEL XORDER
10     ;=====
11     mov eax, HEIGHT
12     mov dword T_HEIGHT, eax
13
14     mov     edx,     SRC
15     mov     edi,     [edx + WIDTH_STEP]
16
17     mov     ecx,     SRC                ;ecx registro para el pixel de origen
18     mov     ecx,     [ecx + IMAGE_DATA]
19
20     mov     esi,     DST                ;esi registro para el pixel de destino
21     mov     esi,     [esi + IMAGE_DATA]
22
23     add     esi,     edi                ;salteo la primera linea
24     inc     esi                        ;salteo la primer columna
25

```

```

26 cicloY:
27     mov edx, edi
28     sub  edx, 2                ;le resto los pixeles laterales
29
30     add  ecx, 2                ;sumo dos para llevar al primero de la linea
31     add  esi, 2                siguiente
32
33     cicloX:
34         mov  eax, [ecx]        ;carga cuatro pixeles en eax
35         and  eax, 0x00FF00FF    ;paso a dos words empaquetadas
36         shr  eax, 8            ;desplazo ocho bits a derecha para permitir
37                                operaciones en 8 bits
38         mov  ebx, [ecx + edi]    ;sumo dos veces la segunda linea
39         and  ebx, 0x00FF00FF    ;paso a dos words empaquetadas
40         shl  ebx, 1            ;desplazo siete bits a derecha para permitir
41                                operaciones en 8 bits
42         add  eax, ebx
43         mov  ebx, [ecx + edi * 2] ;sumo la tercera linea
44         and  ebx, 0x00FF00FF    ;paso a dos words empaquetadas
45         shr  ebx, 8            ;desplazo ocho bits a derecha para permitir
46                                operaciones en 8 bits
47         add  eax, ebx
48         mov  ebx, eax
49         shr  eax, 16            ;muevo eax a la parte izquierda de la matriz
50         sub  ax, bx             ;se la resto al pixel destino
51         cmp  ax, 0x00FF
52         jg   sobresaturado
53         cmp  ax, 0x0000
54         jl   subsaturado
55         volver:
56
57         add  [esi], al          ;mando el pixel
58         inc  ecx
59         inc  esi
60         dec  edx
61         jnz  cicloX
62
63     dec  dword T_HEIGHT
64     jnz  cicloY
65     jmp  pintaBordes
66
67 ySobel:
68
69     cmp  dword YORDER, 0
70     je   pintaBordes

```

### 3.5. Cálculo de complejidad

Hacemos cálculo de complejidad?

Para referenciar una línea es así:

(4)  $O(1) \rightarrow$  ya que es una gilada.

### 3.6. Demostración

Demostramos que el algoritmo realmente hace lo que nos piden?

### 3.7. Análisis

Acá tendríamos que analizar la respuesta de nuestro algoritmo respecto del de opencv.

Un ejemplo de tabla:

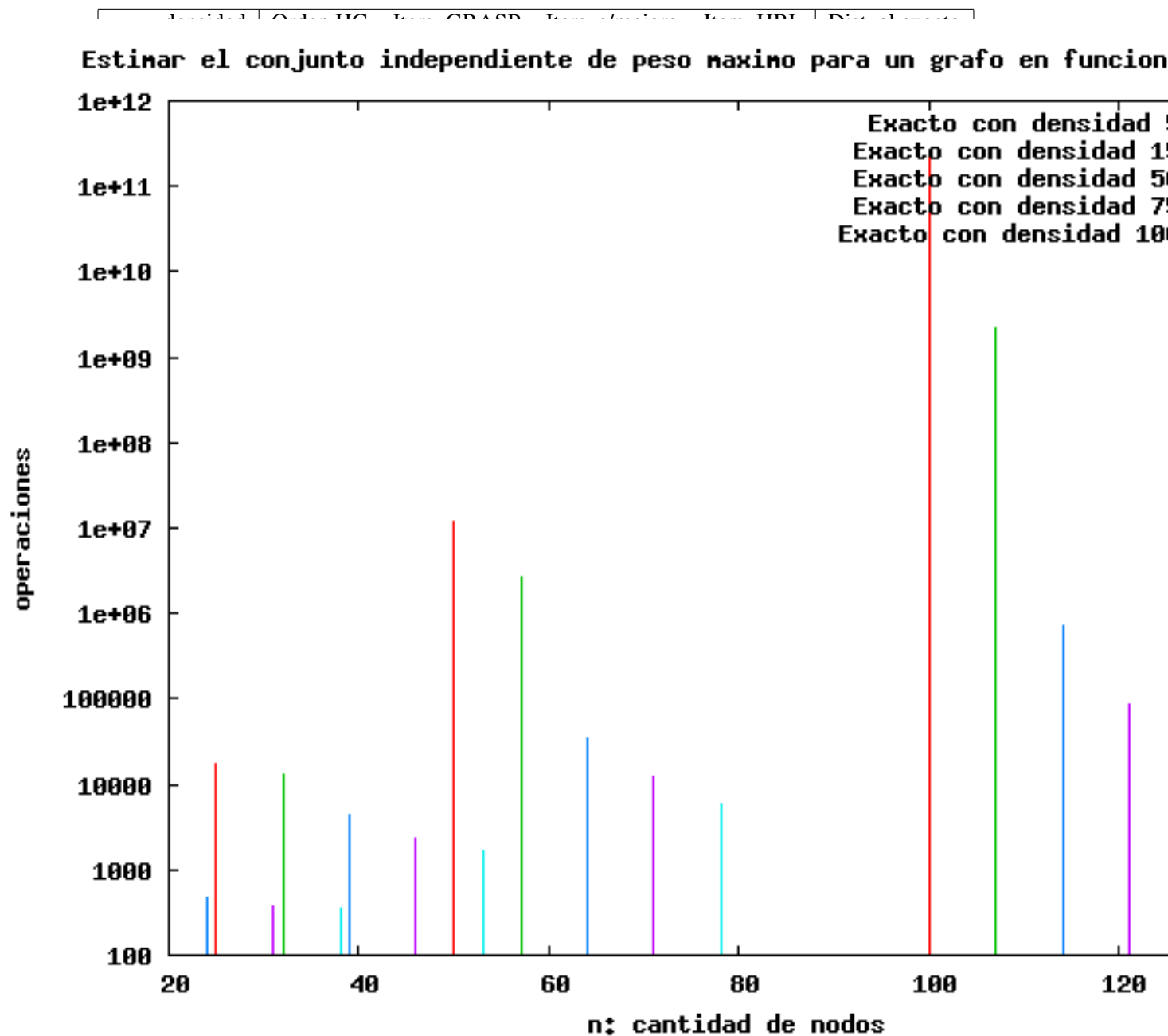


Figura 3.1: Tiempos de ejecución del algoritmo SARASA

## **4. Conclusiones**

Acá metemos las conclusiones del trabajo?