

Proyecto de programación en ensamblador

Estructura de Computadores

- El proyecto consiste en la programación, en ensamblador del Motorola 88110, de un conjunto de rutinas que realicen el *filtrado de una imagen* mediante un filtro programable.
- La imagen será una matriz de píxeles, cada uno de los cuales se representa mediante un *byte sin signo* que especifica su nivel de gris (0 equivale a negro y 255 a blanco).
- El filtro está basado en una operación recursiva de convolución con un núcleo representado por tres constantes, K, C y B, que definen una matriz cuadrada de coeficientes fraccionarios.
- Cada elemento del filtro es en realidad un número fraccionario que se obtiene al dividir el correspondiente elemento de la matriz, C o B, entre la constante K.

Enunciado 2014/2015

- El enunciado de este proyecto está basado en el planteado el semestre/curso anterior, por lo que aquellos alumnos que tengan que repetir o corregir el proyecto que desarrollaron entonces, podrán partir de los programas ya realizados anteriormente, si bien *deberán superar las pruebas que se establezcan para la convocatoria actual*.
- Por otra parte debe observar con atención el apartado de este documento que describe las normas de entrega y, en particular, la especificación del contenido que debe incluir la memoria del proyecto.
- El hecho de plantear un proyecto similar al del curso anterior tiene además las siguientes implicaciones:
 - Se realizará una revisión minuciosa de los proyectos realizados en este semestre para descartar o localizar posibles casos de copia que desafortunadamente se siguen produciendo (y detectando) en la mayoría de las convocatorias.
 - Los alumnos que ya hubieran formado parte de un grupo durante convocatorias anteriores solo podrán establecer grupo con los mismos compañeros que tuvieron en dicho curso o semestre o, alternativamente, realizar el proyecto de forma individual.

Definición del filtro

Filtro es una matriz cuadrada de orden 3 definida por la terna de coeficientes (K, C, B)

Filtro		
B/K	B/K	B/K
B/K	C/K	B/K
B/K	B/K	B/K

Filtro (8, 0, 1)		
0,125	0,125	0,125
0,125	0,0	0,125
0,125	0,0125	0,125

Cálculo del filtro de una imagen (1)

$Im_original = (m[i,j]), \quad Filtro = (f[i,j]), \quad Im_filtrada = (r[i,j])$

$$\begin{aligned} r[i,j] = & f[0,0] \cdot m[i-1,j-1] + f[0,1] \cdot m[i-1,j] + f[0,2] \cdot m[i-1,j+1] + \\ & + f[1,0] \cdot m[i,j-1] + f[1,1] \cdot m[i,j] + f[1,2] \cdot m[i,j+1] + \\ & + f[2,0] \cdot m[i+1,j-1] + f[2,1] \cdot m[i+1,j] + f[2,2] \cdot m[i+1,j+1] \end{aligned}$$

Im_original								Filtro (8, 0, 1)		
1	2	3	4	5	6	7	...	0,125	0,125	0,125
5	6	7	8	9	0	1	...	0,125	0,0	0,125
9	0	1	2	3	4	5	...	0,125	0,125	0,125
3	4	5	6	7	8	9	...			
7	8	9	0	1	2	3	...			
.....								K=8, C=0, B=1		

Valor de $m[2,2]=1$ Valor de $r[2,2]=4,75 \rightarrow 4$

Cálculo del filtro de una imagen (2)

Filtro			Mat		
0,125	0,125	0,125	1	1	1
0,125	0,0	0,125	1	0	1
0,125	0,125	0,125	1	1	1

Subimagen

6	7	8
0	1	2
4	5	6

K = 8, C = 0, B = 1

- Se opera con la matriz Mat definida por C=0 y B=1
- Se divide entre K (instrucción “divs”)

Valor de $r[2,2] = (6+7+8+0+0+2+4+5+6)/8 = 38/8$

Aplicando “divs” da como resultado 4

Filtrado de los bordes de una imagen

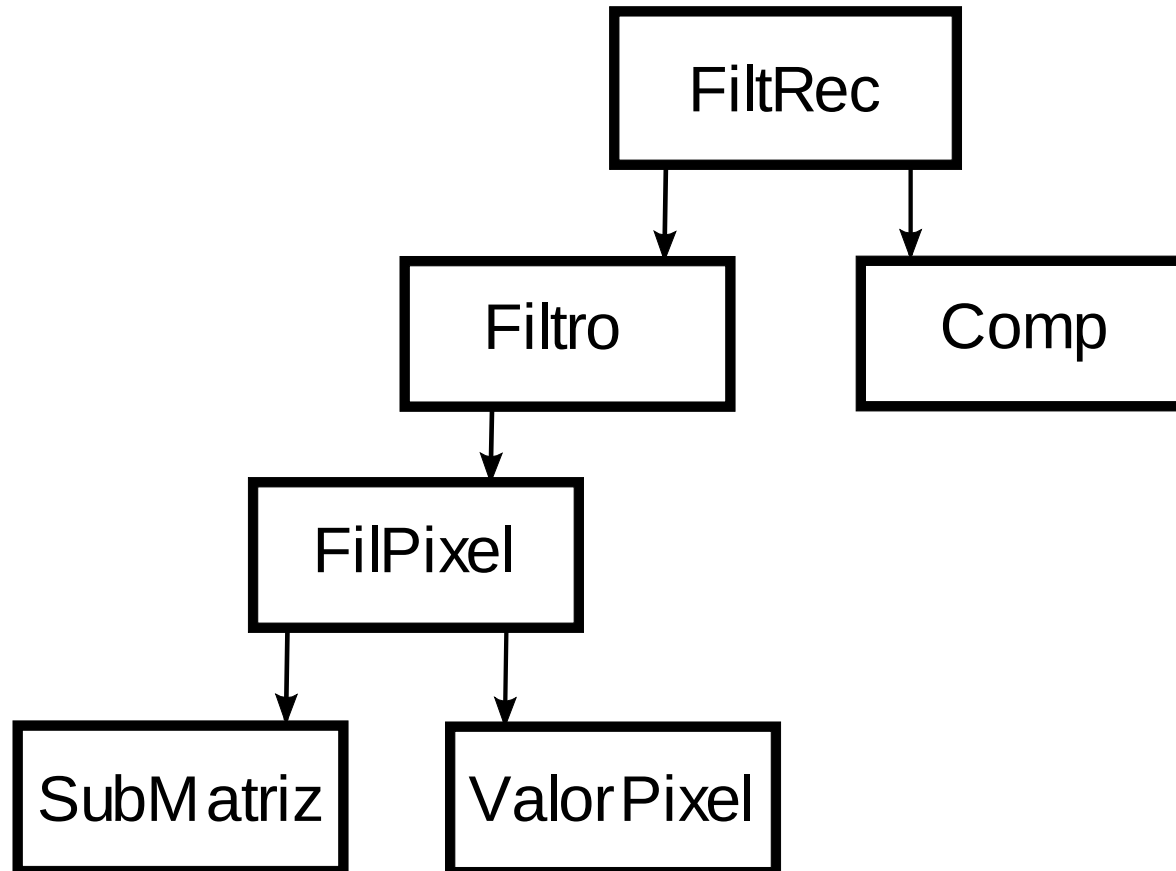
Imagen Original

1	2	3	4	5	6	7	...
5	6	7	8	9	0	1	...
9	0	1	2	3	4	5	...
3	4	5	6	7	8	9	...
7	8	9	0	1	2	3	...
...	...	...	...	...	...	...	...

Subimágenes de los bordes

1	1	1					
1	1	1	3	4	6	6	6
1	1	1	7	8	6	6	6
	9	0	1	2	3	4	5
	3	4	5	6	7	8	9
	7	8	9	0	1	2	3
	...	...	...	...	...	...	...

Jerarquía de las rutinas



Compara imágenes

Diferencias = Comp (Imagen1, Imagen2)

- Parámetros (en la pila):
 - Imagen1:** De entrada. Se pasa por dirección: M (1 palabra), N (1 palabra), MxN elementos (1 byte cada uno).
 - Imagen2:** De entrada. Se pasa por dirección: M (1 palabra), N (1 palabra), MxN elementos (1 byte cada uno).
- Valor de retorno:
 - Diferencias:** Devuelve en r29 el valor de la diferencia entre las imágenes.
- Algoritmo:
 1. Inicializar a cero un acumulador de diferencias (*Dif*).
 2. Recorrer los MxN píxeles de cada matriz incrementando Dif en el valor absoluto de la diferencia: $Dif = Dif + \text{abs}(Im1_{ij} - Im2_{ij})$.
 3. Retornar cargando Dif en r29.

Valor del píxel filtrado (1)

VPixel = ValorPixel (SubImg, MFiltro)

- Parámetros (en la pila):
 - SubImg:** De entrada. Se pasa por dirección: 9 elementos (1 byte sin signo cada uno). Es la submatriz de 3x3 centrada en el píxel a filtrar.
 - MFiltro:** De entrada. Se pasa por dirección. Formada por:
 - La constante K (entero de una palabra) que divide a cada componente del filtro
 - El valor C (entero de una palabra) del elemento central del filtro
 - El valor B (entero de una palabra) de los 8 elementos del borde del filtro
- Valor de retorno:
 - VPixel:** Devuelve en r29 el valor (entero sin signo en el rango 0..255) del píxel filtrado.
- Descripción:
 1. Aplica el filtro MFiltro a la submatriz SubImg.

Valor del píxel filtrado (2)

VPixel = ValorPixel (SubImg, MFiltro)

- Algoritmo
 1. Inicializa a cero el acumulador ACC y lee los valores K, C y B.
 2. Inicializa un puntero al primer elemento de SubImg.
 3. Recorre, en un bucle, los 4 primeros elementos de SubImg multiplicándolos por el valor B. En cada iteración:
 - Se pasa el valor del píxel sin signo a un registro **rxx**
 - Se multiplica **rxx** por el coeficiente B del filtro y se acumula en ACC
 4. Se lee el valor central de SubImg, se multiplica por el coeficiente C y se acumula en ACC
 5.

Valor del píxel filtrado (3)

VPixel = ValorPixel (SubImg, MFiltro)

- Algoritmo (Continuación...)
 5. Recorre, en un bucle, los 4 últimos elementos de SubImg multiplicándolos por el valor B. En cada iteración:
 - Se pasa el valor del píxel sin signo a un registro **rxx**
 - Se multiplica **rxx** por el coeficiente B del filtro y se acumula en ACC
 6. Se divide ACC entre K, dejando el resultado en ACC. Usará la división entera.
 7. Si $ACC < 0$ se ajusta a 0 y si $ACC > 255$ se ajusta a 255.
 8. Se pone $r29 = ACC$ y se retorna al llamante.

Extracción de Submatriz (1)

SubMatriz (Imagen, SubImg, i, j)

- Parámetros (en la pila):
 - Imagen:** De entrada. Se pasa por dirección: M (1 palabra), N (1 palabra), MxN elementos (1 byte cada uno). Es la imagen de la que se debe extraer una submatriz de 3x3 elementos.
 - SubImg:** De salida. Se pasa por dirección: 9 elementos (1 byte sin signo cada uno). Es la matriz de 3x3 elementos en la que se debe copiar la submatriz de Imagen centrada en el pixel [i, j].
 - i:** Número de fila. (0..M-1). De entrada. Se pasa por valor.
 - j:** Número de columna. (0..N-1). De entrada. Se pasa por valor.
- Descripción:
 1. Obtiene la submatriz SubImg de 3x3 elementos centrada en el píxel [i, j] de la Imagen.

Extracción de Submatriz (2)

SubMatriz (Imagen, SubImg, i, j)

- Algoritmo
 1. Determinar si el píxel $[i, j]$ pertenece al borde de la Imagen, es decir, si $i=0$ o $j=0$ o $i=M-1$ o $j=N-1$.
 2. Si pertenece al borde: Copiar el píxel $[i, j]$ sobre los 9 píxeles de SubImg.
 3. Si no pertenece al borde: Copiar la submatriz de Imagen centrada en $[i, j]$ sobre los 9 píxeles de SubImg.

Filtro de un píxel (1)

VPixel = FilPixel (Imagen, i, j, MFiltro)

- Parámetros (en la pila):

Imagen: De entrada. Se pasa por dirección: M (1 palabra), N (1 palabra), MxN elementos (1 byte cada uno). Es la imagen que contiene el píxel a filtrar.

i: Número de fila. De entrada. Se pasa por valor.

j: Número de columna. De entrada. Se pasa por valor.

MFiltro: De entrada. Se pasa por dirección. Formada por:

- La constante K (entero de una palabra) que divide a cada componente del filtro
- El valor C (entero de una palabra) del elemento central del filtro
- El valor B (entero de una palabra) de los 8 elementos del borde del filtro

- Valor de retorno:

VPixel: Devuelve en r29 el valor del píxel filtrado.

- Descripción:

1. Aplica la máscara de filtrado al píxel [i,j] de la imagen.

Filtro de un píxel (2)

VPixel = FilPixel (Imagen, i, j, MFiltro)

- Algoritmo
 1. Reserva espacio en el marco de pila para almacenar una submatriz de 3x3 bytes sin signo, (SubImg), (3 palabras).
 2. Llama a SubMatriz (Imagen, SubImg, i, j).
 3. Llama a ValorPixel (SubImg, Mfiltro).
 4. Retorna al llamante teniendo r29 el valor devuelto por ValorPixel.

Filtro de una imagen (1)

Filtro (Imagen, ImFiltrada, MFiltro)

- Parámetros (en la pila):
 - Imagen:** De entrada. Se pasa por dirección: M (1 palabra), N (1 palabra), MxN elementos (1 byte cada uno). Es la imagen que se debe filtrar.
 - ImFiltrada:** De salida. Es la imagen que resulta de aplicar el filtro a la imagen de entrada. Se pasa por dirección: M (1 palabra), N (1 palabra), MxN elementos (1 byte cada uno).
 - MFiltro:** De entrada. Se pasa por dirección. Formada por:
 - La constante K (entero de una palabra) que divide a cada componente del filtro
 - El valor C (entero de una palabra) del elemento central del filtro
 - El valor B (entero de una palabra) de los 8 elementos del borde del filtro
- Descripción:
 - Aplica la máscara de filtrado definida por MFiltro a la imagen de entrada. Deja el resultado en la dirección definida por ImFiltrada.

Filtro de una imagen (2)

Filtro (Imagen, ImFiltrada, MFiltro)

- Algoritmo:
 1. Copiar M y N sobre la imagen filtrada.
 2. Desde $i=0$ hasta $i=M-1$
 - Desde $j=0$ hasta $j=N-1$
 1. Preparar parámetros y llamar a `FilPixel (Imagen, i, j, MFiltro)`.
 2. Almacenar el valor de retorno en la posición $[i,j]$ de la imagen filtrada.
 3. Retornar al llamante.

Filtro recursivo (1)

NFiltrados=FiltRec(Imagen, MFiltro, NCambios, NFiltrados)

- Parámetros (en la pila):

Imagen: De entrada y salida. Se pasa por dirección: M (1 palabra), N (1 palabra), MxN elementos (1 byte cada uno). Es la imagen de entrada que se recibe y la imagen de salida una vez filtrada.

MFiltro: De entrada. Se pasa por dirección. Formada por:

- La constante K (entero de una palabra) que divide a cada componente del filtro
- El valor C (entero de una palabra) del elemento central del filtro
- El valor B (entero de una palabra) de los 8 elementos del borde del filtro

NCambios: De entrada y se pasa por valor (1 palabra). Es el valor de la diferencia entre la imagen original y la filtrada que determina si se debe hacer o no una nueva llamada recursiva.

NFiltrados: De entrada y se pasa por valor (1 palabra). Es el número de operaciones de filtrado ya realizadas.

Filtro recursivo (2)

NFiltrados=FiltRec(Imagen, MFiltro, NCambios, NFiltrados)

- Valor de retorno:

NFiltrados: Devuelve en r29 el número de veces que se ha aplicado el filtro a la imagen original.

- Condición de salida de la recursividad:

1. NFiltrados ≥ 15 (valor límite prefijado). El Programa principal llamante ha de pasar el parámetro NFiltrados con valor 0.
2. La diferencia entre la imagen original y la filtrada sea **menor** que NCambios.

Filtro recursivo (3)

NFiltrados=FiltRec(Imagen, MFiltro, NCambios, NFiltrados)

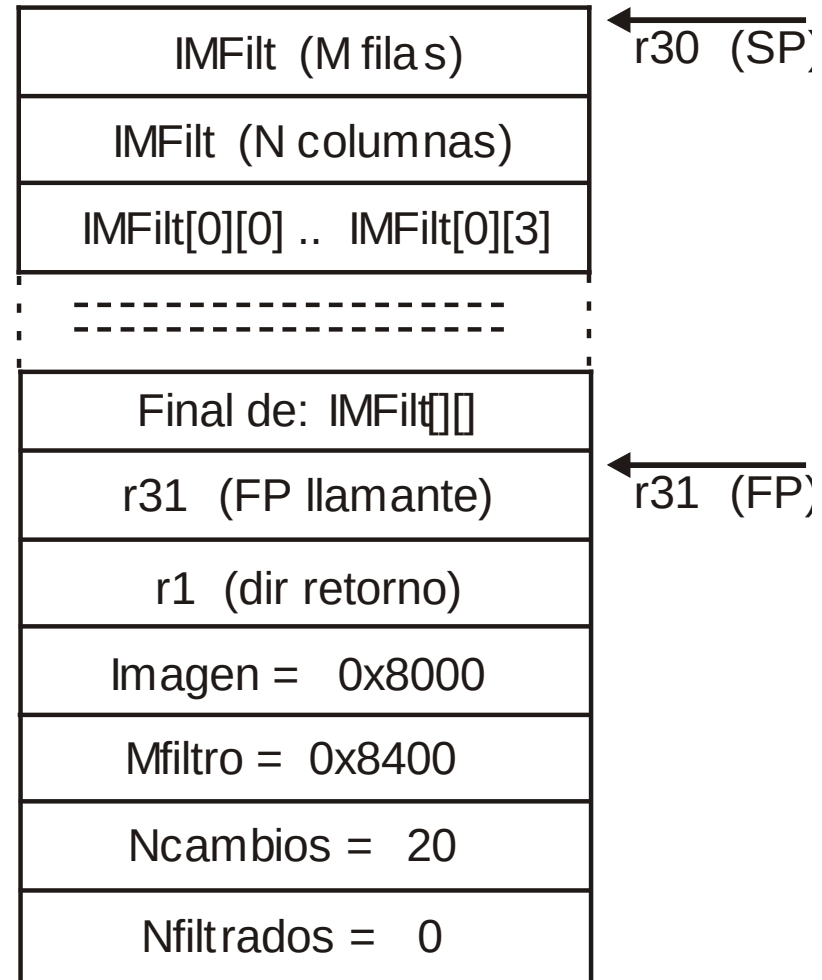
- Algoritmo:
 1. Reservar espacio en el marco de pila para almacenar la imagen filtrada: $4+4+M \cdot N$ ajustado por exceso a múltiplo de 4.
 2. Llamar a Filtro (Imagen, ImFiltrada, MFiltro) con ImFiltrada con la dirección anterior.
 3. Incrementar NFiltrados. Si es $\geq 15 \rightarrow$ Copiar NFiltrados en r29 y continuar en paso 6.
 4. Llamar a Comp pasando las imágenes original y filtrada. Si $r29 < Ncambios \rightarrow$ Copiar NFiltrados en r29 y continuar en paso 6.
 5. Llamar a FiltRec pasando la imagen filtrada (ImFiltrada), MFiltro, NCambios y el valor incrementado NFiltrados del punto 3. (Al retornar de esta llamada recursiva continúa en paso 6. El valor devuelto en r29 es el número de filtrados finalmente realizados. La imagen filtrada se habrá actualizado con el resultado de la última operación de filtrado)
 6. Copiar la imagen filtrada (que está en el marco de pila) en la zona que ocupaba la original (indicada por el parámetro Imagen).
 7. Retornar al llamante con $r29 = \text{NFiltrados}$.

Tratamiento de la pila

FiltRec:

```

PUSH    (r1)
PUSH    (r31)
addu    r31, r30, r0
ld      r10, r31, 8
.....
add     r6, r6, 3
clr     r6, r6, 2<0>
subu    r30, r30, r6
    
```



Comparación de dos imágenes. Caso 1.

r30=36856 (0x8FF8)

Direcciones de memoria:

36848			00800000	28800000
--------------	--	--	-----------------	-----------------

32768	04000000	08000000	00000000	00000000
--------------	-----------------	-----------------	-----------------	-----------------

32784	00000000	00070000	00000000	00000000
--------------	-----------------	-----------------	-----------------	-----------------

32800	00000000	00000000		
--------------	-----------------	-----------------	--	--

32800			04000000	08000000
--------------	--	--	-----------------	-----------------

32816	00000000	00000000	00000000	00000000
--------------	-----------------	-----------------	-----------------	-----------------

32832	00000000	00000000	00000000	00000000
--------------	-----------------	-----------------	-----------------	-----------------

Resultado:

r30=36856 (0x8FF8) r29=7 (0x7)

Extracción de submatriz. Caso 3.

r30=36848 (0x8FF0)

Direcciones de memoria:

36848 00800000 40800000 01000000 01000000

32768 03000000 03000000 10203040 50607080

32784 90000000

32832 FFFFFFFF FFFFFFFF FFFFFFFF

Resultado:

r30=36848 (0x8FF0)

Direcciones de memoria:

32832 10203040 50607080 90FFFFFF

Valor del píxel filtrado. Caso 5.

r30=36856 (0x8FF8)

Direcciones de memoria:

36848	00800000	10800000
--------------	-----------------	-----------------

32768	00000000	55000000	00000000
--------------	-----------------	-----------------	-----------------

32784	01000000	01000000	00000000
--------------	-----------------	-----------------	-----------------

Resultado:

r30=36856 (0x8FF8) r29=85 (0x55)

Filtro de un píxel. Caso 8.

r30=36848 (0x8FF0)

Direcciones de memoria:

36848	00800000	02000000	02000000	30800000
--------------	-----------------	-----------------	-----------------	-----------------

32768	04000000	08000000	44444444	44444444
--------------	-----------------	-----------------	-----------------	-----------------

32784	44353433	44444444	44448844	44444444
--------------	-----------------	-----------------	-----------------	-----------------

32800	44444444	44444444		
--------------	-----------------	-----------------	--	--

32816	08000000	00000000	01000000	
--------------	-----------------	-----------------	-----------------	--

Resultado:

r30=36848 (0x8FF0) r29=62 (0x3E)

Filtro de un píxel. Caso 8.

Imagen:

44	44	44	44	44	44	44	44
44	35	34	33	44	44	44	44
44	44	88	44	44	44	44	44
44	44	44	44	44	44	44	44

MFiltro (K=8, C=0, B=1):

1	1	1	
1	0	1	K=8
1	1	1	

Filtro de una imagen. Caso 10.

r30=36852 (0x8FF4)

Direcciones de memoria:

36848	00800000	28800000	50800000
--------------	-----------------	-----------------	-----------------

32768	04000000	08000000	01020304	04050607
--------------	-----------------	-----------------	-----------------	-----------------

32784	11421314	14851617	21222324	74252627
--------------	-----------------	-----------------	-----------------	-----------------

32800	31323334	34353637		
--------------	-----------------	-----------------	--	--

32800			00000000	00000000
--------------	--	--	-----------------	-----------------

32816	00000000	00000000	00000000	00000000
--------------	-----------------	-----------------	-----------------	-----------------

32832	00000000	00000000	00000000	00000000
--------------	-----------------	-----------------	-----------------	-----------------

32848	01000000	04000000	00000000	
--------------	-----------------	-----------------	-----------------	--

Filtro de una imagen. Caso 10.

Resultado:

r30=36852 (0x8FF4)

Direcciones de memoria:

32800			04000000	08000000
32816	04080C10	1014181C	44FF4C50	50FF585C
32832	84888C90	FF94989C	C4C8CCD0	D0D4D8DC

Filtro recursivo. Caso 12.

r30=36848 (0x8FF0)

Direcciones de memoria:

36848	00800000	30800000	00000000	00000000
--------------	-----------------	-----------------	-----------------	-----------------

32768	04000000	08000000	FF0000FF	FF0000FF
--------------	-----------------	-----------------	-----------------	-----------------

32784	FF0000FF	FF0000FF	FF0000FF	FF0000FF
--------------	-----------------	-----------------	-----------------	-----------------

32800	FF0000FF	FF0000FF		
--------------	-----------------	-----------------	--	--

32816	08000000	00000000	01000000	
--------------	-----------------	-----------------	-----------------	--

Resultado:

r30=36848 (0x8FF0) r29=15 (0x0f)

Direcciones de memoria

32768	04000000	08000000	FF0000FF	FF0000FF
--------------	-----------------	-----------------	-----------------	-----------------

32784	FF8F7795	95778FFF	FF8F7795	95778FFF
--------------	-----------------	-----------------	-----------------	-----------------

32800	FF0000FF	FF0000FF		
--------------	-----------------	-----------------	--	--

Filtro recursivo. Caso 12.

Ini:

FF	00	00	FF	FF	00	00	FF
FF	00	00	FF	FF	00	00	FF
FF	00	00	FF	FF	00	00	FF
FF	00	00	FF	FF	00	00	FF

Result:

FF	00	00	FF	FF	00	00	FF
FF	8F	77	95	95	77	8F	FF
FF	8F	77	95	95	77	8F	FF
FF	00	00	FF	FF	00	00	FF

r29=15 (0x0f)

Entrega (fechas)

CONVOCATORIA DE JUNIO 2015

El plazo de entrega del proyecto estará abierto desde el:

viernes día 27 de marzo hasta el lunes 4 de mayo de 2015

en que se realizará la corrección definitiva. A partir del día 6 de mayo el sistema estará configurado de modo que permita entregar únicamente la memoria del proyecto. El plazo de entrega para dicha memoria finalizará el viernes 8 de mayo a las 21:00.

Cada grupo podrá disponer de **cinco** de las correcciones del proyecto planificadas para los días 8 a 10, 13 a 17, 20 a 24 y 27 a 30 de abril. Además de esas cinco, podrá utilizar las correcciones de los días **27 de marzo y 7 de abril**, así como la última, prevista para el día **4 de mayo**.

Todas las correcciones se realizarán a partir de las 21:00. Para solicitar una corrección bastará con entregar correctamente los ficheros del proyecto antes de dicha hora límite.

El examen del proyecto está planificado para el viernes día 8 de mayo en el horario de evaluación habilitado por Jefatura de Estudios.

Entrega (contenido)

1. *Memoria*, en formato DIN A4, en cuya portada deberá figurar claramente el nombre y apellidos de los autores del proyecto, identificador del grupo de alumnos (el mismo que emplean para realizar las entregas y consultas) y el nombre de la asignatura. Dicha memoria se entregará en formato electrónico *PDF*.
2. *Ficheros*:
 1. **autores**: Es un fichero ASCII que deberá contener los apellidos, nombre, número de matrícula, DNI y dirección de correo electrónico de los autores del proyecto. El proyecto se realizará individualmente o en grupos de dos alumnos. Cada línea de este fichero contendrá los datos de uno de los autores de acuerdo al siguiente formato:
No Matrícula; DNI ; apellido apellido, nombre; correo electrónico
 2. **filtror.ens**: Contendrá las subrutinas que componen el proyecto junto con un programa principal que se haya utilizado para su depuración.
 3. **memoria.pdf**: fichero conteniendo la memoria del proyecto.

Entrega (memoria)

La memoria debe contener los siguientes puntos:

- Histórico del desarrollo de las rutinas, con fechas, avances y dificultades encontradas, especificando el trabajo que realiza cada miembro del grupo o si dicho trabajo es común. Se detallará en este apartado:
 - Número total de horas invertidas en el proyecto por cada miembro del grupo.
 - Relación de visitas realizadas a los profesores del proyecto.
- Descripción resumida del juego de ensayo (conjunto de casos de prueba) que el grupo haya diseñado y utilizado para probar el correcto funcionamiento del proyecto.
- Observaciones finales y comentarios personales de este proyecto, entre los que se debe incluir una descripción de las principales dificultades surgidas para su realización.