



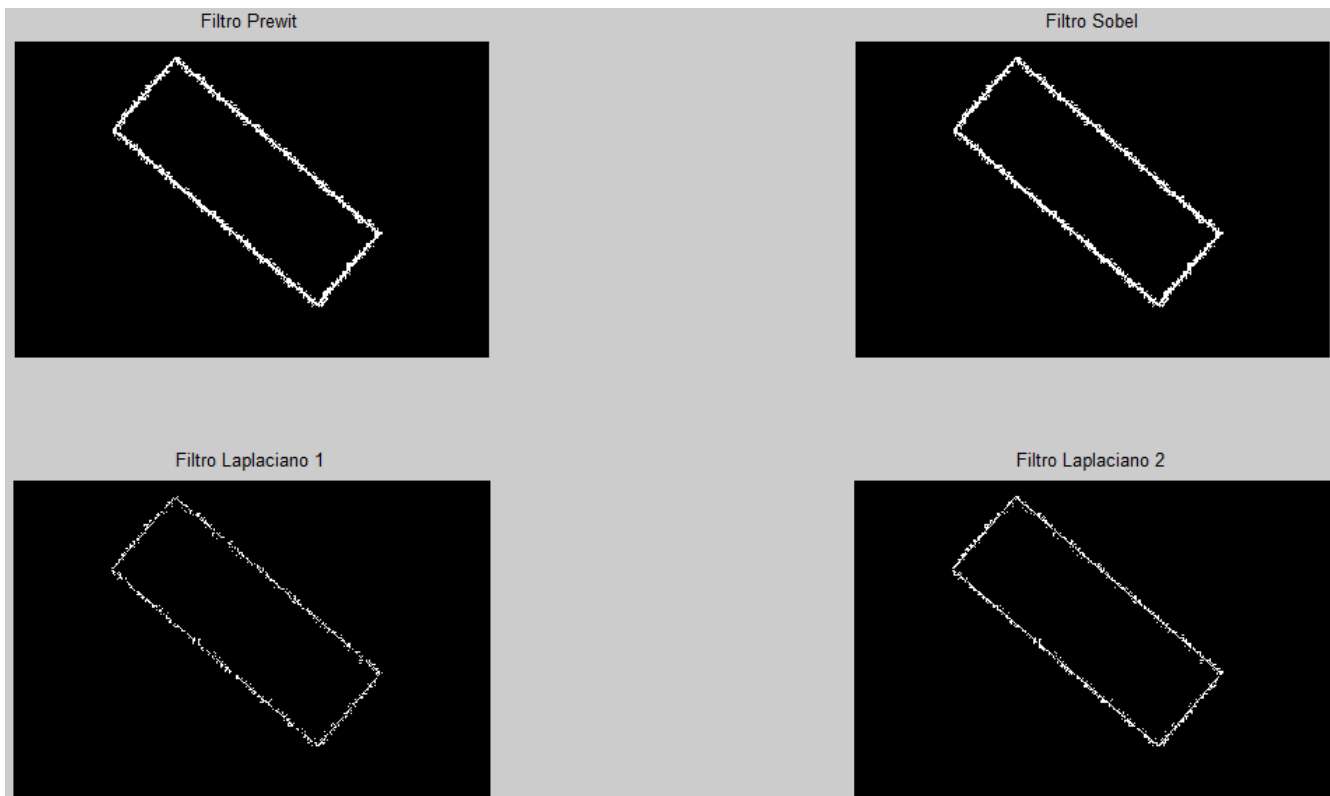
## OTROS FILTROS

**Script:** (ver anexo con las funciones utilizadas)

```
foto=double(rgb2gray(imread('rectangulo2.jpg')));  
A=prewitt(foto);  
B=sobel(foto);  
E1=[0,-1,0;-1,4,-1;0,-1,0];  
E2=[-1,-1,-1;-1,8,-1;-1,-1,-1];  
C=filtro3x3(foto,E1);  
D=filtro3x3(foto,E2);
```

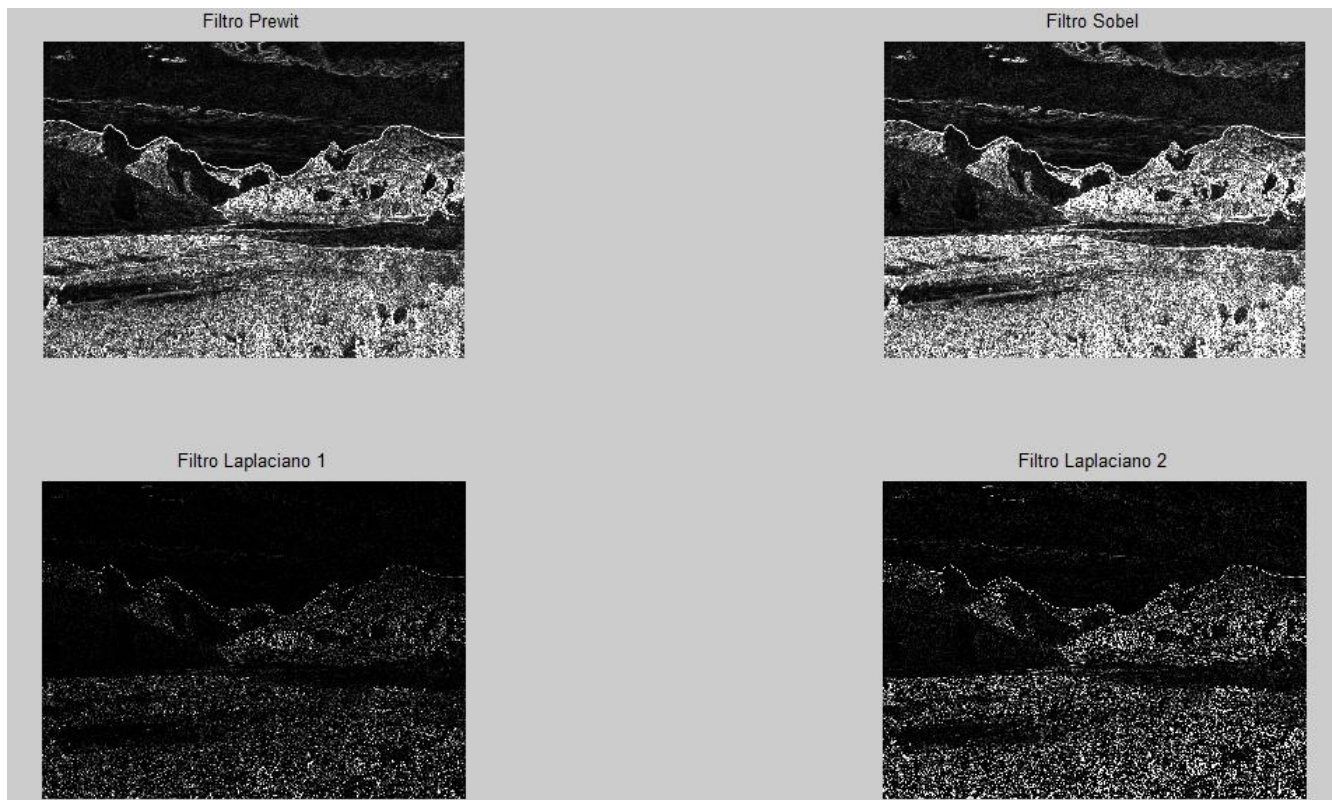
```
figure  
subplot(2,2,1)  
    imshow(A)  
    title('Filtro Prewit')  
subplot(2,2,2)  
    imshow(B)  
    title('Filtro Sobel')  
subplot(2,2,3)  
    imshow(C)  
    title('Filtro Laplaciano 1')  
subplot(2,2,4)  
    imshow(D)  
    title('Filtro Laplaciano 2')
```

### Resultados con rectángulo:





### Resultados con foto:



### Filtro de dilatación y erosión:

Ya que los filtros mínimo y máximo actúan como filtros de dilatación y erosión en imágenes binarias, no se crearon nuevas funciones para este ejercicio solo un pequeño script en cada caso.

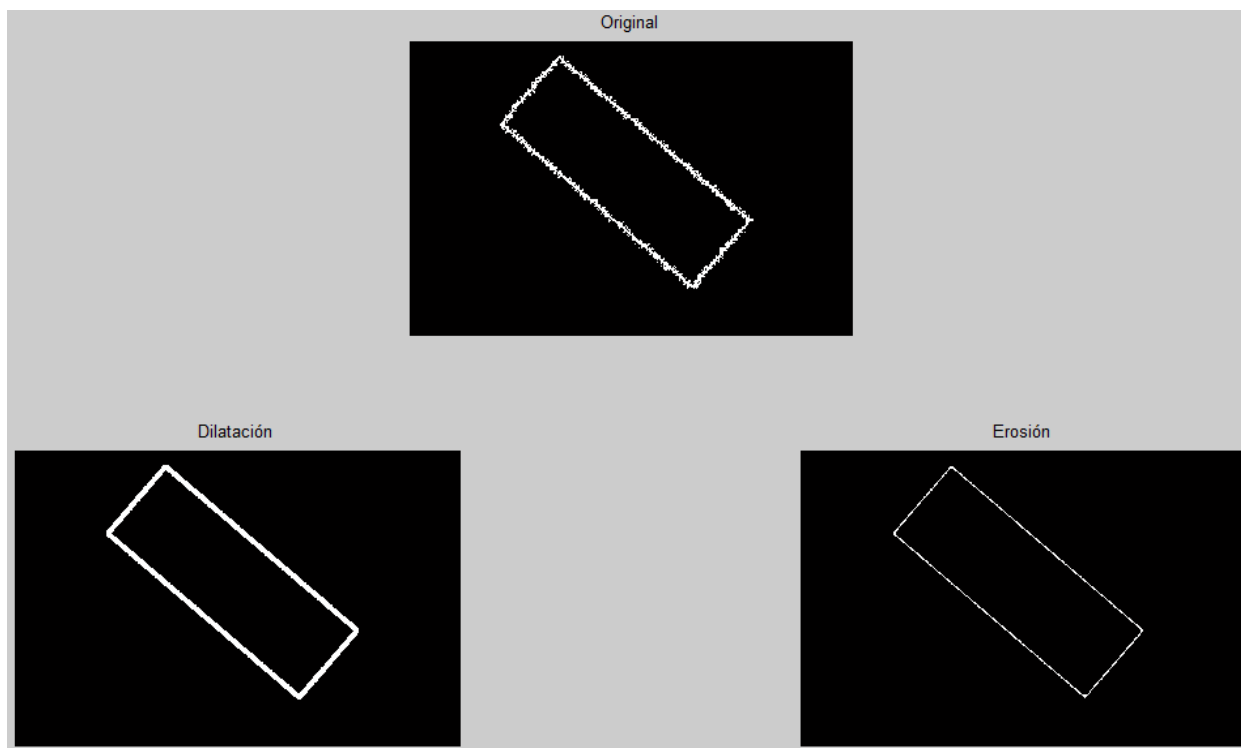
### Dilatación y erosión con rectángulo previamente filtrado:

```
E=filtromax(binarizar(A,200),7); %A es la imagen con el filtro de Prewitt.  
F=filtromin(E,7); %Erosión de la imagen previamente dilatada.
```

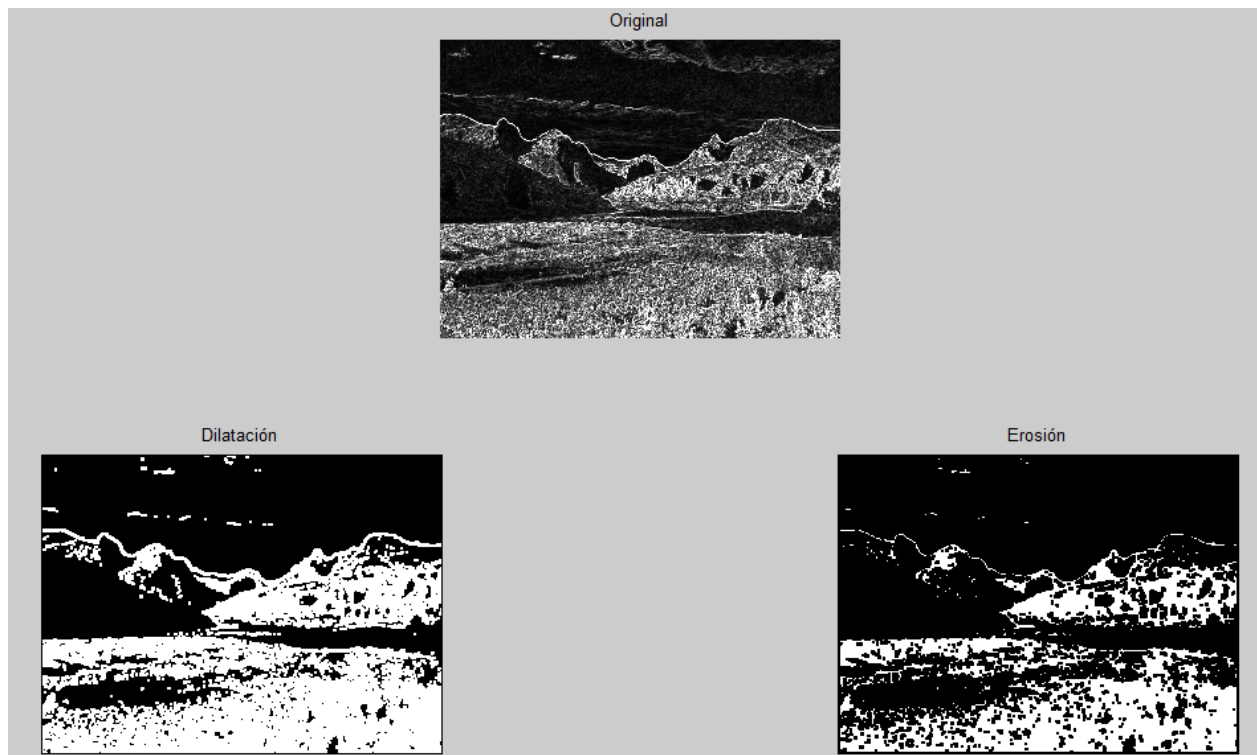
```
figure  
subplot(2,1,1)  
    imshow(A)  
    title('Original')  
subplot(2,2,3)  
    imshow(E)  
    title('Dilatación')  
subplot(2,2,4)  
    imshow(F)  
    title('Erosión')
```



**Resultado:**



**Dilatación y erosión con foto previamente filtrada:**





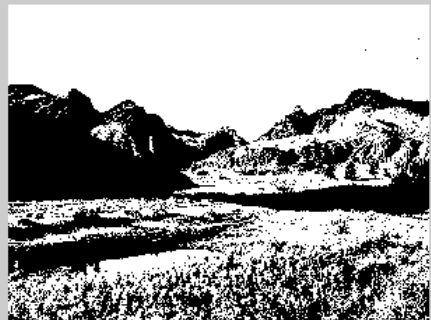
### Dilatación y erosión con foto:

```
G=binarizar(foto,100);  
H=filtromax(foto,5)  
I=filtromin(foto,13)  
  
figure  
subplot(2,2,1)  
    imshow(foto)  
    title('Original')  
subplot(2,2,2)  
    imshow(G)  
    title('Foto binarizada')  
subplot(2,2,3)  
    imshow(H)  
    title('Dilatación')  
subplot(2,2,3)  
    imshow(I)  
    title('Erosión')
```

Original



Foto binarizada



Dilatación



Erosión





## ANEXO: Funciones utilizadas.

```
function [ S ] = prewitt( E )
EE1 = [1,1,1;0,0,0;-1,-1,-1];
EE2 = rot90(EE1);
EE3 = rot90(EE2);
EE4 = rot90(EE3);
A=filtro3x3(E,EE1);
B=filtro3x3(E,EE2);
C=filtro3x3(E,EE3);
D=filtro3x3(E,EE4);
S=abs(A+B+C+D);
end
```

```
function [ S ] = sobel( E )
EE1 = [1,2,1;0,0,0;-1,-2,-1];
EE2 = rot90(EE1);
EE3 = rot90(EE2);
EE4 = rot90(EE3);
A=filtro3x3(E,EE1);
B=filtro3x3(E,EE2);
C=filtro3x3(E,EE3);
D=filtro3x3(E,EE4);
S=abs(A+B+C+D);
end
```

```
function [ S ] = filtro3x3(E,EE)
%E es la matriz de entrada y EE el elemento estructurante de 3x3.
[a,b]=size(E);
S=zeros(a,b);
for i=2:a-1
    for j=2:b-1
        S(i,j) = sum(sum(E(i-1:i+1,j-1:j+1).*EE));
        if (S(i,j)<0) S(i,j)=0; end
        if (S(i,j)>255) S(i,j)=255; end
    end
end
end
```

```
function [ S ] = filtromin( E , n)
%E es la matriz de entrada y n es el tamaño de la matriz cuadrada
[a,b]=size(E);
S = zeros(a,b);
if (mod(n,2)==0) %Para n par
    for i=1+n/2:a-n/2-1
        for j=1+n/2:b-n/2-1
            S(i,j) = min(min(E(i-n/2:i+n/2-1,j-n/2:j+n/2-1)));
        end
    end
else %Para n impar
    for i=1+(n-1)/2:a-(n-1)/2
        for j=1+(n-1)/2:b-(n-1)/2
            S(i,j) = min(min(E(i-(n-1)/2:i+(n-1)/2,j-(n-1)/2:j+(n-1)/2)));
        end
    end
end
end
end
```

```
function [ S ] = filtromax( E , n)
%E es la matriz de entrada y n es el tamaño de la matriz cuadrada
[a,b]=size(E);
S = zeros(a,b);
if (mod(n,2)==0) %Para n par
    for i=1+n/2:a-n/2-1
        for j=1+n/2:b-n/2-1
            S(i,j) = max(max(E(i-n/2:i+n/2-1,j-n/2:j+n/2-1)));
        end
    end
else %Para n impar
    for i=1+(n-1)/2:a-(n-1)/2
        for j=1+(n-1)/2:b-(n-1)/2
            S(i,j) = max(max(E(i-(n-1)/2:i+(n-1)/2,j-(n-1)/2:j+(n-1)/2)));
        end
    end
end
end
end
```