



## Tratamiento de Señales

Version 2024-I

# Operaciones Morfológicas

[ Capítulo 7 ]

**Dr. José Ramón Iglesias**

DSP-ASIC BUILDER GROUP

Director Semillero TRIAC

Ingenieria Electronica

Universidad Popular del Cesar

# Morfología:

Se utilizan para manipular la forma de imágenes

## Morfología:

Se utilizan para manipular la forma de imágenes principalmente binarias, aunque muchas de sus operaciones se pueden usar en imágenes en tonos de gris.

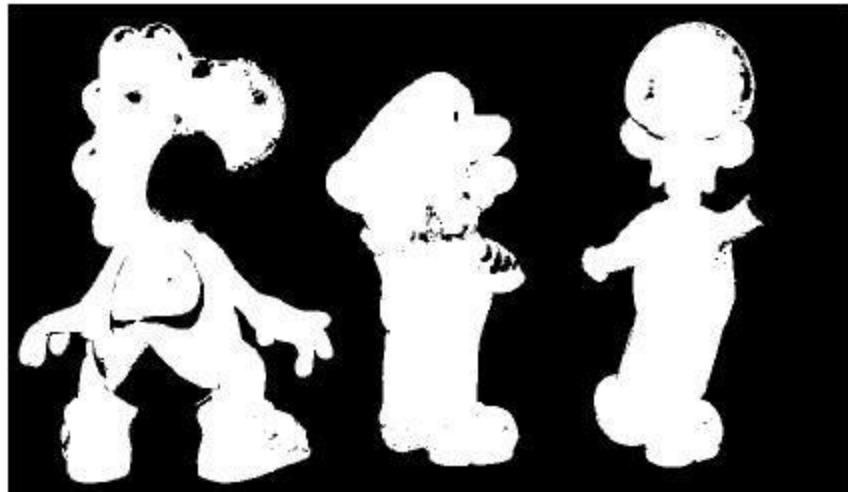
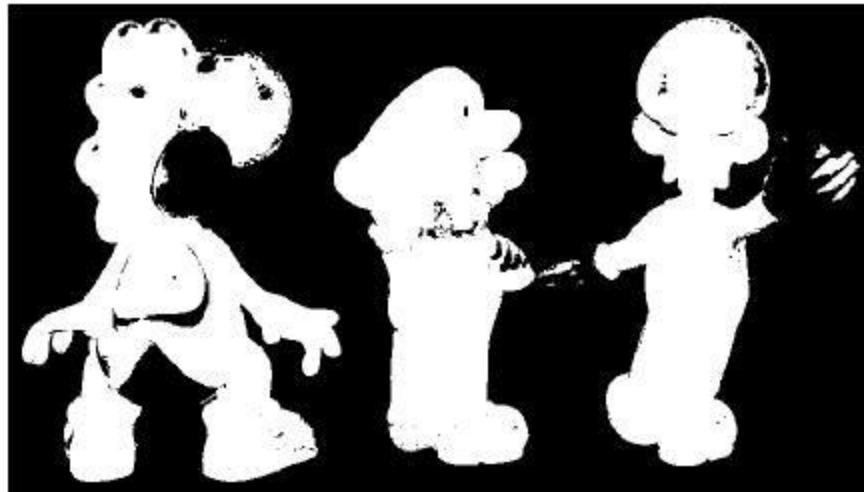
## Reducción de ruido en una imagen binaria

Al binarizar una imagen pueden aparecer:

- Píxeles aislados que no pertenecen a ningún objeto de interés, estos deben ser **eliminados**.
- Huecos dentro de un objeto, los cuales deben ser **rellenados**.

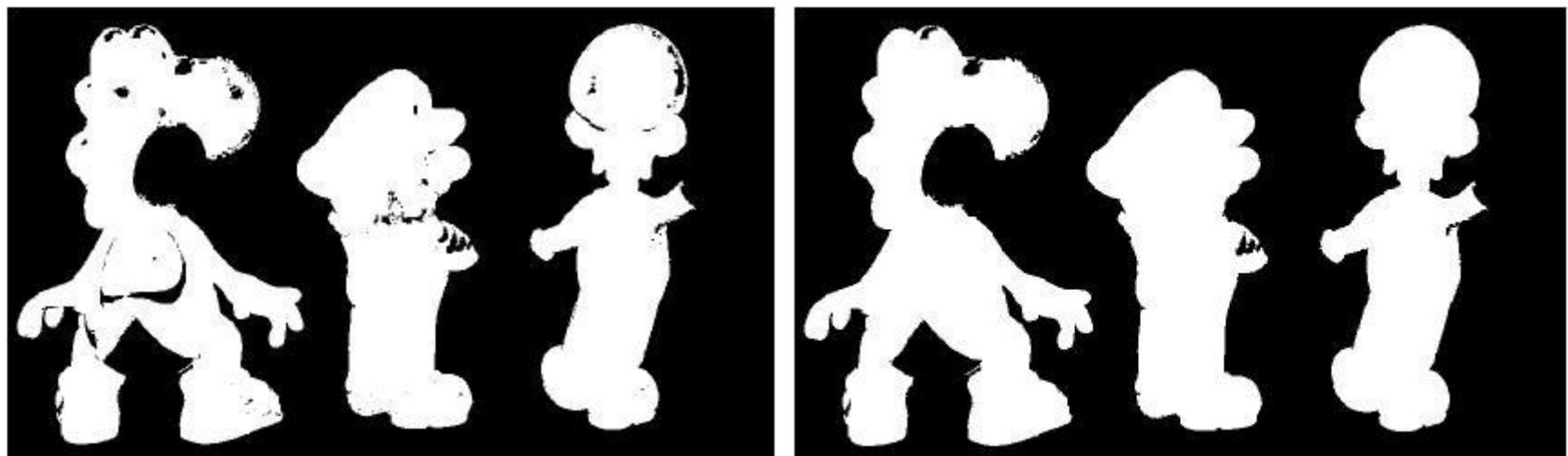
**Idea:** Eliminar los objetos que tengan un área pequeña.

**Ejemplo.** Se eliminan los objetos con área menor a 500 píxeles.



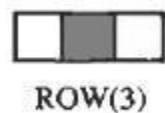
Idea: Rellenar los huecos negros que se encuentre rodeados por píxeles blancos.

En matlab: imfill('BW',holes);

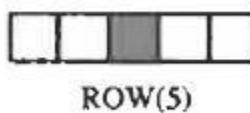


## Operaciones morfológicas

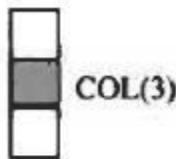
Se realizan sobre cada píxel de la imagen binaria teniendo en cuenta sus vecinos. Para tal fin, la imagen es ventaneada.



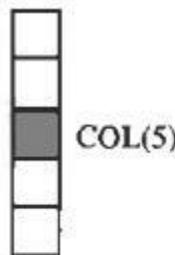
ROW(3)



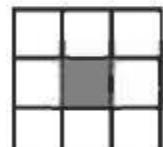
ROW(5)



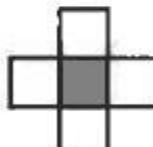
COL(3)



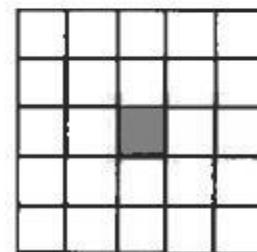
COL(5)



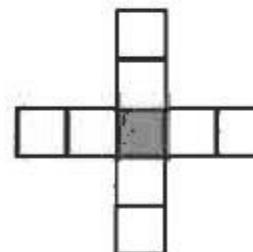
SQUARE(9)



CROSS(5)



SQUARE(25)



CROSS(9)

Algunas ventanas típicas, con centro en  $(m,n)$ .

Ejemplo. Algunos ventaneos.

- Para la ventana ROW(3)

$$W[I(m,n)] = [I(m,n-1) \quad I(m,n) \quad I(m-1,n+1)]$$

- Para la ventana SQUARE(9)

$$W[I(m,n)] = \begin{bmatrix} I(m-1,n-1) & I(m-1,n) & I(m-1,n+1) \\ I(m,n-1) & I(m,n) & I(m,n+1) \\ I(m+1,n-1) & I(m+1,n) & I(m+1,n+1) \end{bmatrix}$$

**Erosión.** Se realiza una AND entre todo los píxeles de la ventana.

$$\text{Ero}(m,n) = \text{AND}\{W[I(m,n)]\} = \min\{W[I(m,n)]\}$$

Características:

tomar el valor mínimo de la imagen en el entorno de vecindad definido por el elemento estructurante

- Expande el fondo
- Suaviza la frontera del objeto
- Remueve los objetos más pequeños que la ventana

Ejemplo. Erosión en Python: erosion = cv2.erode(img,kernel,iterations = 1)

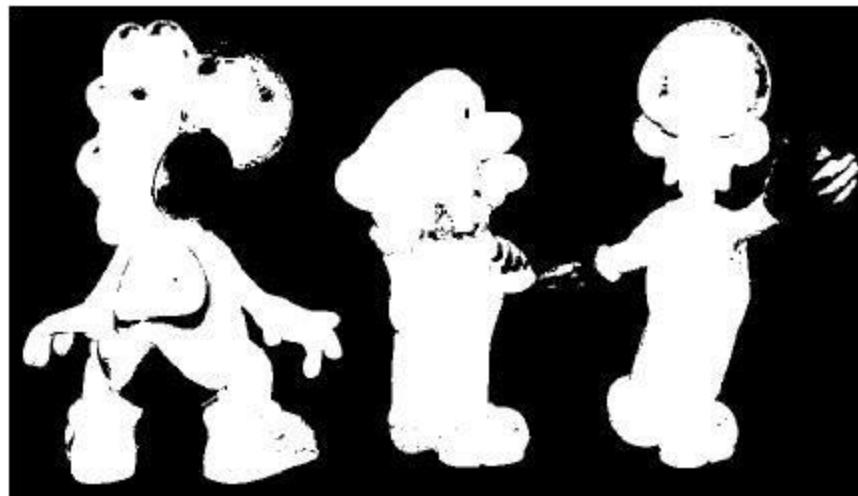
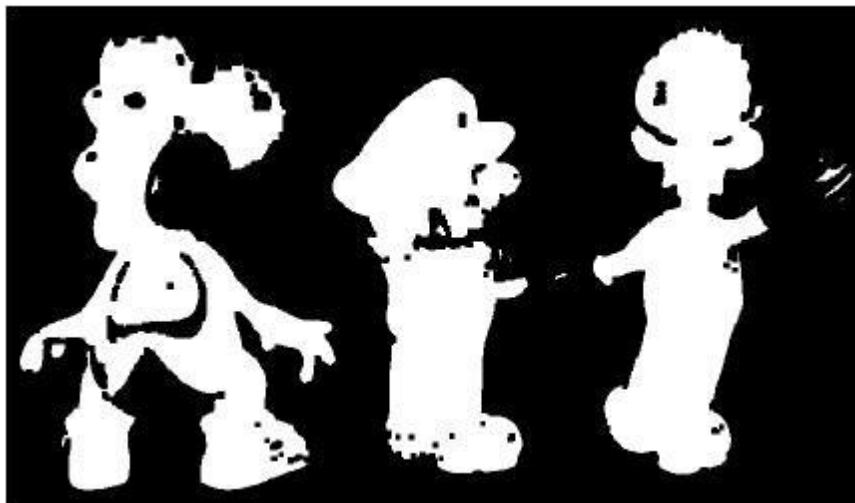
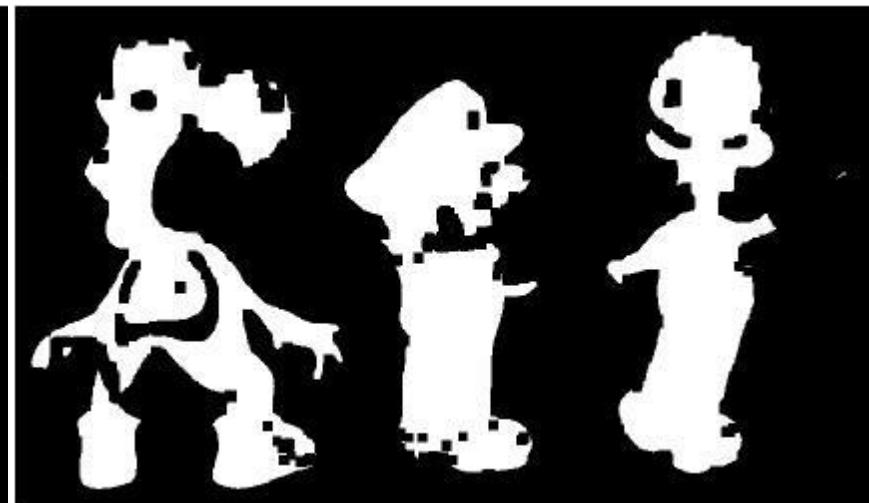


Imagen original



Erosionada con SQUARE 5x5



Erosionada con SQUARE 9x9

**Dilatación.** Se realiza una OR entre todo los píxeles de la ventana y el resultado obtenido es puesto en el píxel central.

$$\text{Dil}(m,n) = \text{OR} \{W[I(m,n)]\} = \max \{W[I(m,n)]\}$$

Características:

- Expande el tamaño del objeto
- Suaviza la frontera del objeto
- Remueve los huecos más pequeños que la ventana

La dilatación también se interpreta como el valor máximo del entorno de vecindad definido por el elemento estructurante

Ejemplo. Erosión en Python: dilation = cv2.dilate(img,kernel,iterations = 1)

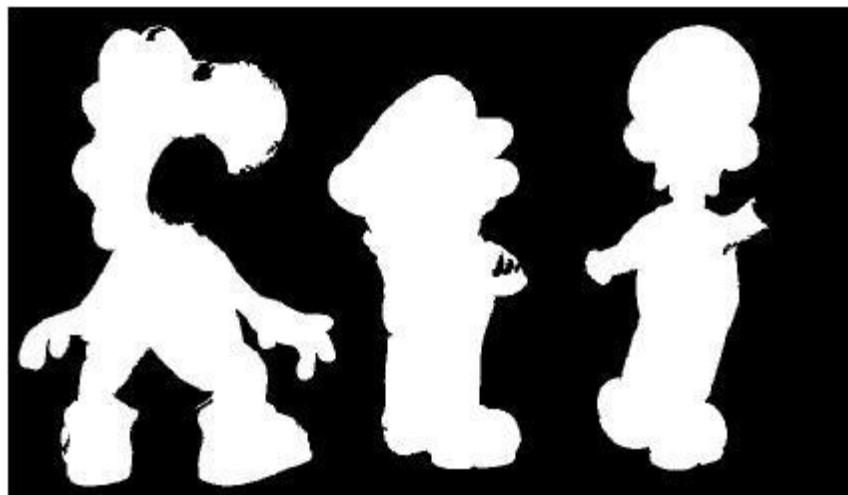
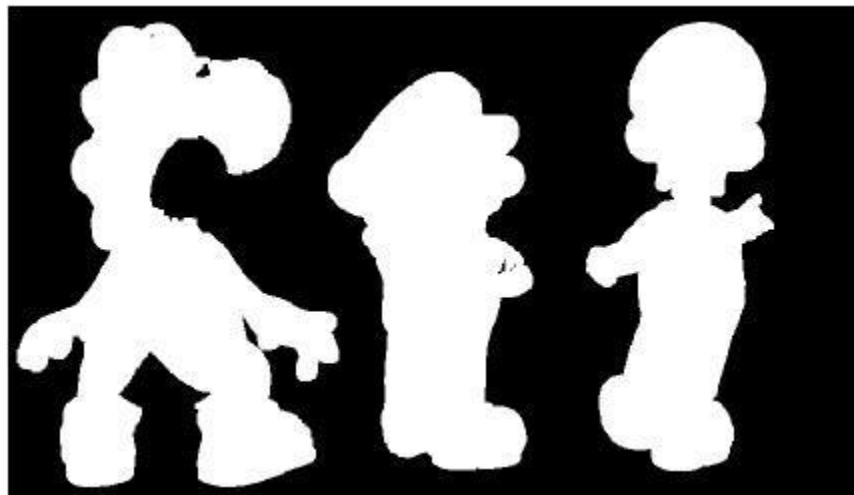
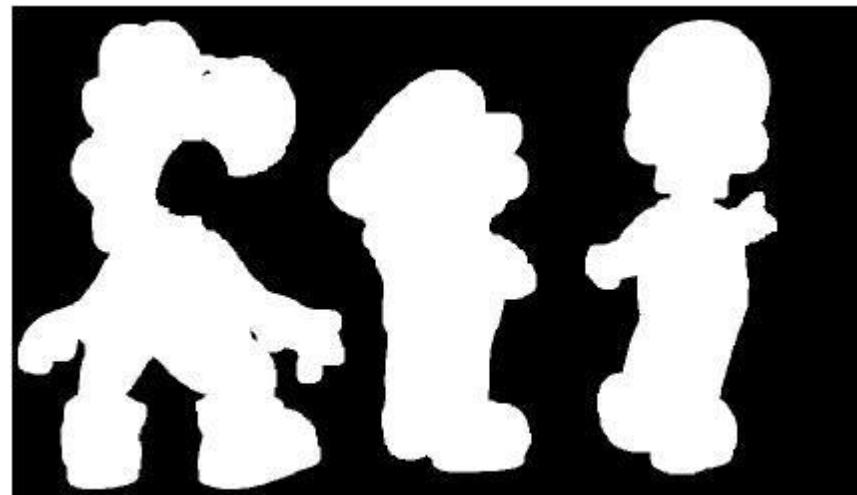


Imagen original



Dilatada con SQUARE 5x5



Dilatada con SQUARE 9x9

Eliminando el fondo detectado de la imagen original, se tiene



**Opening.** Permite **eliminar objetos pequeños** (ruido) sin distorsionar los objetos más grandes.

$$\text{Open}(m,n) = \text{Dil}\{\text{Ero}\{I(m,n)\}\}$$

**Closing.** Permite **rellenar pequeños huecos** de los objetos sin distorsionar los mismos.

$$\text{Close}(m,n) = \text{Ero}\{\text{Dil}\{I(m,n)\}\}$$



Original



Opening



Closing



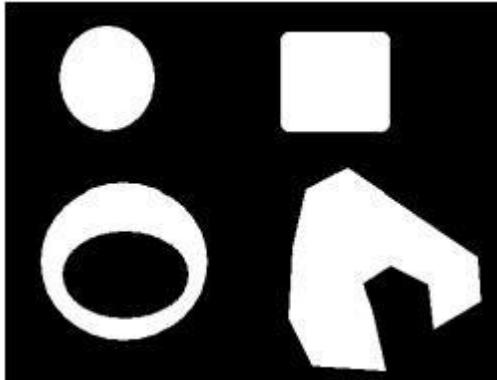
Closing + Opening

## Contorno o Frontera de un objeto

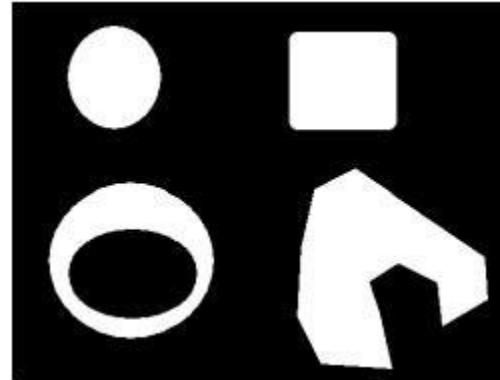
El contorno de un objeto binario puede ser encontrado por medio de operaciones morfológicas.

$$\text{Erosión: } I_c = I - \text{Ero}(I)$$

Imagen

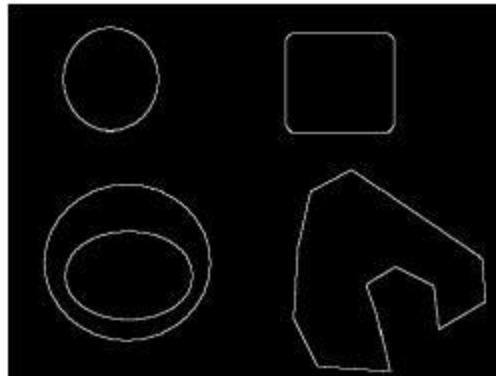


Erosión



Contorno

=



## Descripción de un contorno

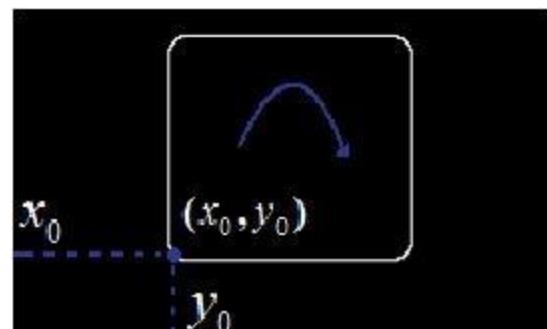
Obtener **descriptores del contorno** de un objeto es útil para reconocer un objeto por su forma automáticamente.

Ejemplo. Triangulo? Rectángulo? Circulo?



## Descriptores de Fourier

Sea  $\{(x_0, y_0), (x_1, y_1), \dots, (x_{N-1}, y_{N-1})\}$  los  $N$  píxeles que pertenecen al contorno del objeto, recorridos en sentido horario.



Una secuencia de  $N$  complejos puede ser definida por medio de los píxeles del contorno, de acuerdo a

$$s[n] = x_n + jy_n, \quad n = 0, 1, \dots, N-1$$

Los **descriptores de Fourier** pueden ser encontrados como

$$c[k] = \frac{1}{N} \sum_{n=0}^{N-1} s[n] e^{-j \frac{2\pi k}{N} n}, \quad k = 0, 1, 2, \dots, N-1$$

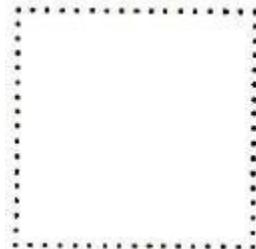
El contorno puede ser recuperado a través de los descriptores

$$s[n] = \sum_{k=0}^{N-1} c[k] e^{j \frac{2\pi k}{N} n}, \quad n = 0, 1, 2, \dots, N-1$$

**Propiedad.** Los  $c[k]$  coeficientes guardan información de los detalles del contorno:

- Los primeros coeficientes (*bajas frecuencias*): detalles grueso
- Los últimos coeficientes (*altas frecuencias*): detalles finos

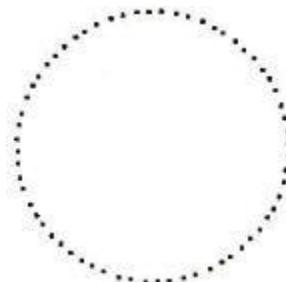
**Ejemplo.** Considere el contorno de la figura



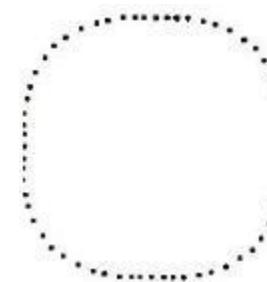
$$N = 64 \Rightarrow 64 \text{ descriptores}$$

Si el contorno es aproximado utilizando sólo los primeros  $M$  descriptores, entonces

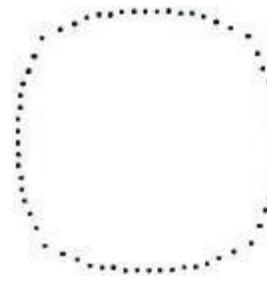
$$\hat{s}[n] = \sum_{k=0}^{M-1} c[k] e^{j \frac{2\pi k}{N} n}, \quad n = 0, 1, 2, \dots, N-1$$



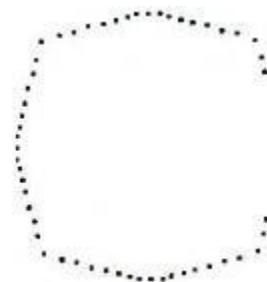
$$M = 2$$



$$M = 8$$



$$M = 32$$

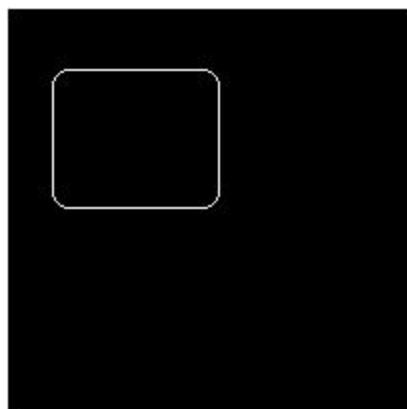


$$M = 61$$

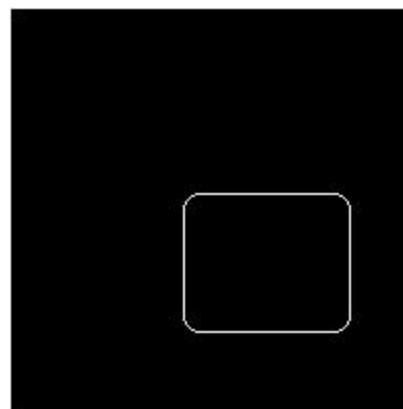
**Observación.** Los descriptores de la forma de un objeto debe ser invariantes a: *traslación* y *escalamiento*

*Efecto de la **traslación** sobre los descriptores*

$s[n]$



$s_T[n]$



$$s_T[n] = s[n] + \Delta_{xy} = (x_n + \Delta_x) + j(y_n + \Delta_y) \Leftrightarrow c_T[k] = c[k] + \Delta_{xy} \delta[k]$$

Sólo se afecta el coeficiente  $c[0]$ !

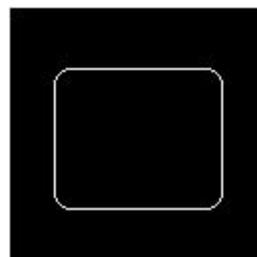
El descriptor  $c[0]$  contiene la información del centro de masa del contorno, pues

$$c[0] = \frac{1}{N} \sum_{n=0}^{N-1} s[n] = \frac{1}{N} \sum_{n=0}^{N-1} x_n + j \frac{1}{N} \sum_{n=0}^{N-1} y_n$$

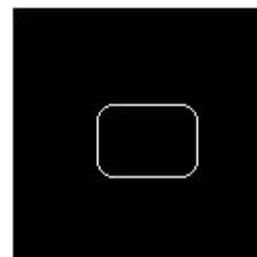
**Observación.**  $c[0]$  no da información sobre la forma del contorno, por tal motivo **se descarta**.

Efecto del **escalamiento** sobre los descriptores

$$s[n]$$



$$s_E[n]$$



$$s_E[n] = \alpha s[n] \Leftrightarrow c_E[k] = \alpha c[k]$$

Todos los coeficientes se escalan por  $\alpha$  !

El descriptor  $c[1]$  contiene la información del tamaño del contorno.

Si todos los descriptores son llevados cero, excepto  $c[1]$  entonces se obtiene una circunferencia

$$\hat{s}[n] = c[1]e^{j\frac{2\pi}{N}n}, \quad n = 0, 1, 2, \dots, N$$

donde  $c[1]$  corresponde al radio de la circunferencia.

**Observación.**  $c[1]$  es utilizado para normalizar los demás coeficientes, de acuerdo a:

$$c_N[k] = \frac{c[k]}{|c[1]|}, \quad k = 2, 3, \dots, N-1$$

## Operaciones morfológicas en Niveles de gris

Sea  $I(m,n)$  una imagen en niveles de gris, entonces

$$\text{Dil}(m,n) = \max\{W[I(m,n)]\}$$



Imagen Original



Imagen Dilatada

## Operaciones morfológicas en Niveles de gris

Sea  $I(m,n)$  una imagen en niveles de gris, entonces

$$\text{Ero}(m,n) = \min \{W[I(m,n)]\}$$



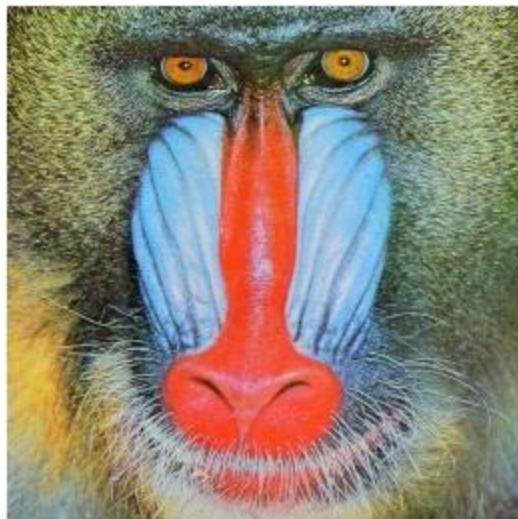
Imagen Original



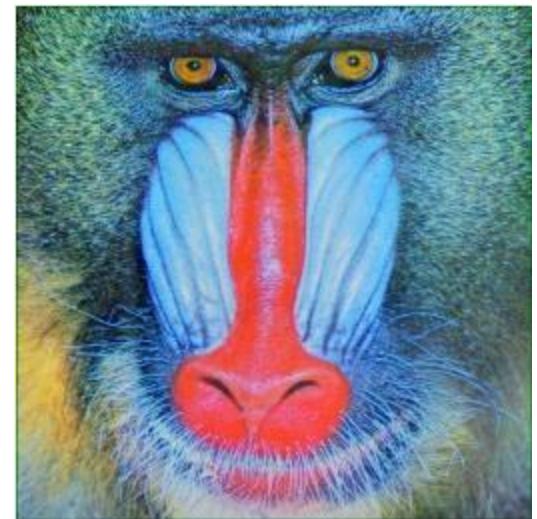
Imagen Erosionada

## Operaciones morfológicas por componentes de color

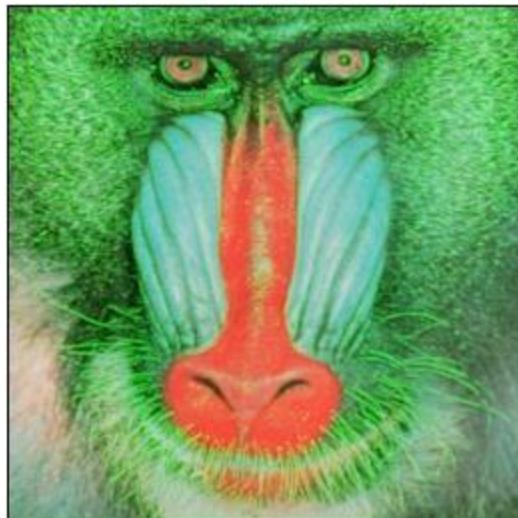
Original



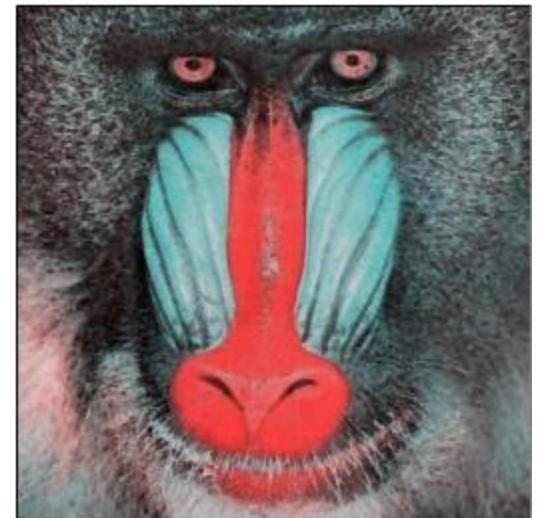
**Ero** en Rojo  
**Dil** en Azul



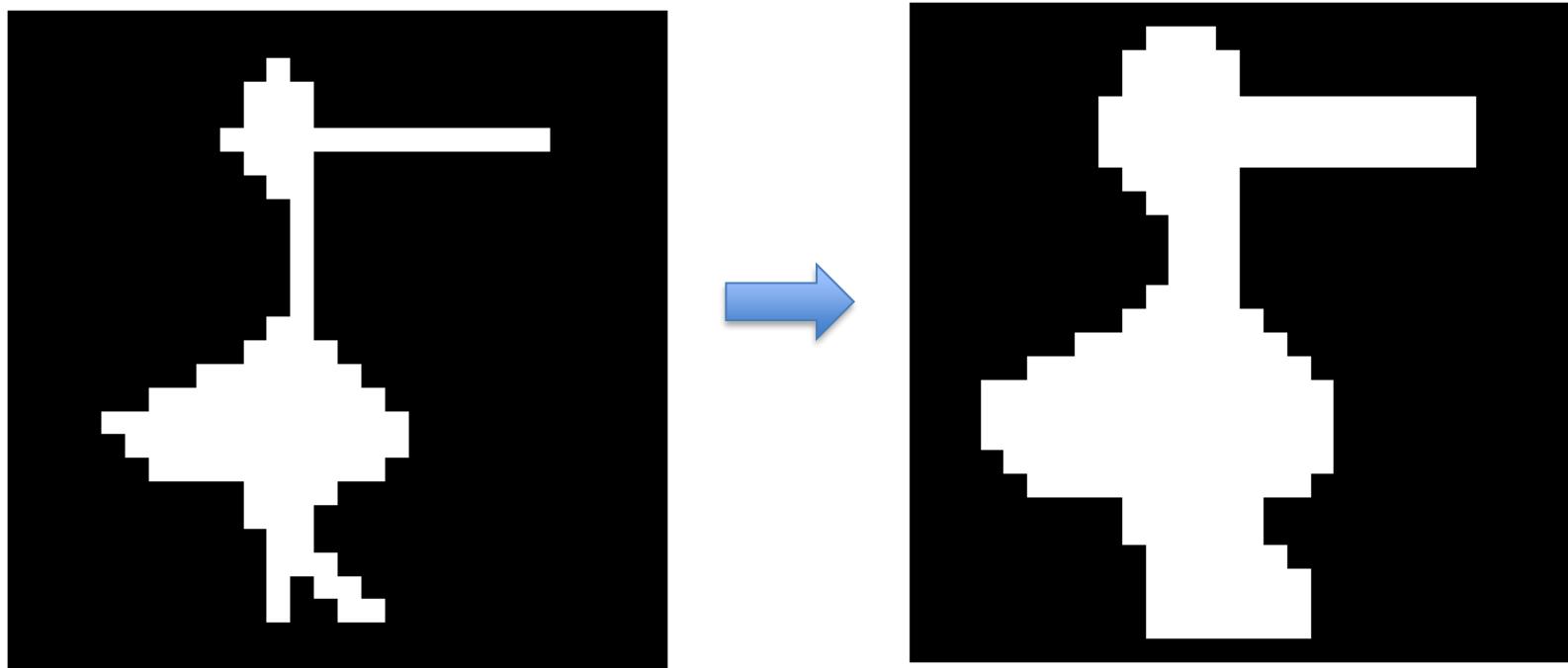
**Dil** en Verde  
**Ero** en Rojo  
y Azul



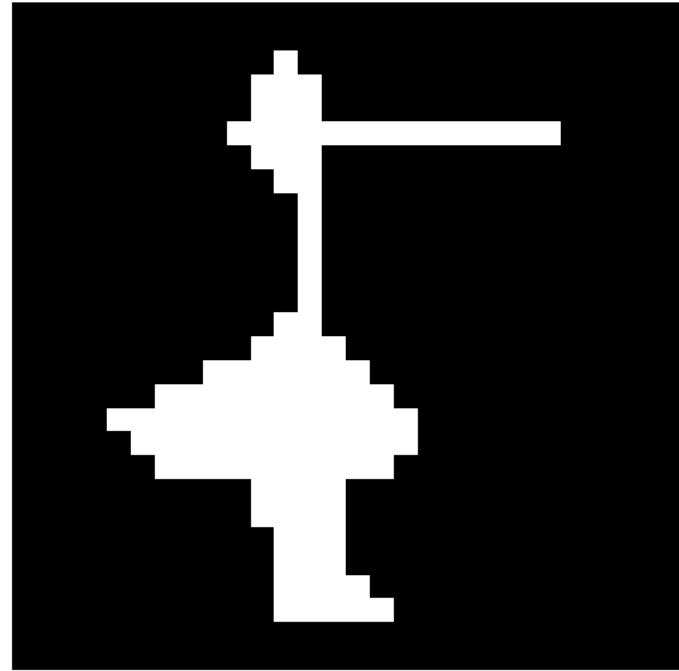
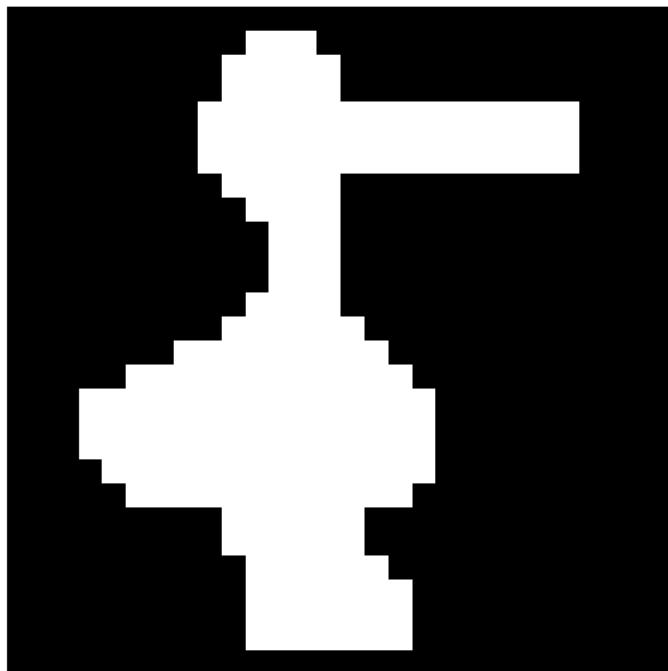
**Ero** en  
R,G,B



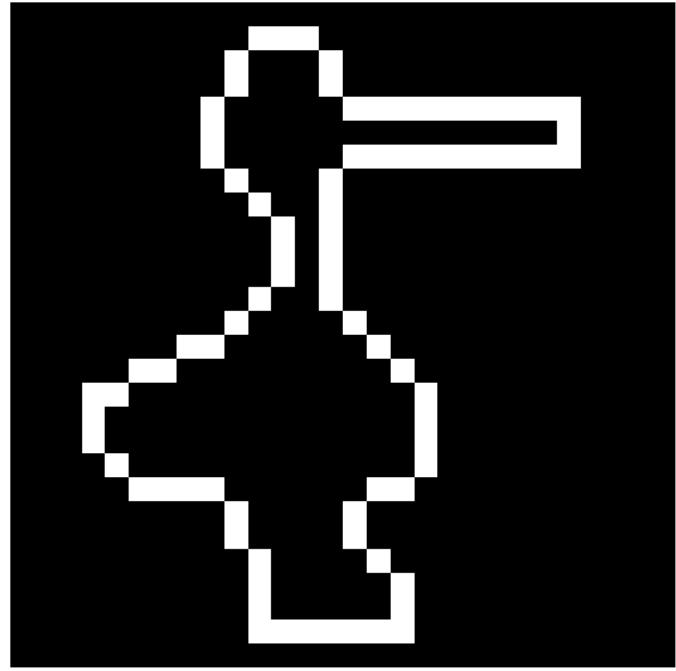
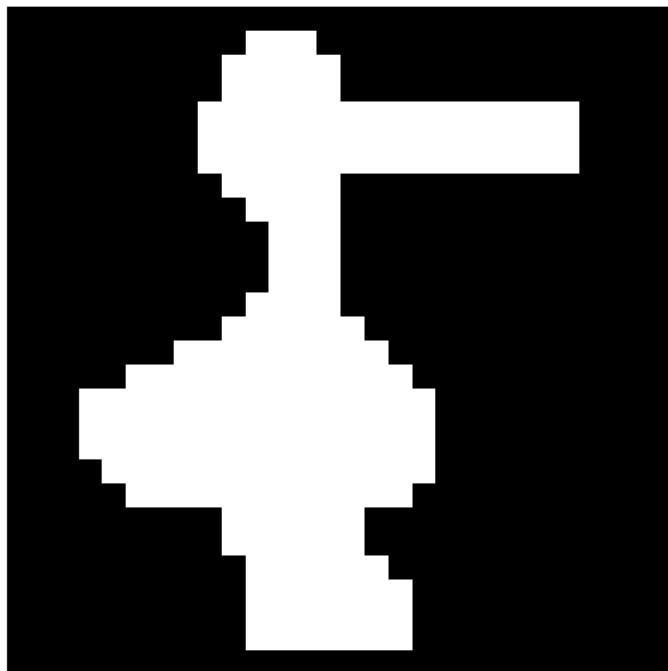
# Ejemplos: Dilatación



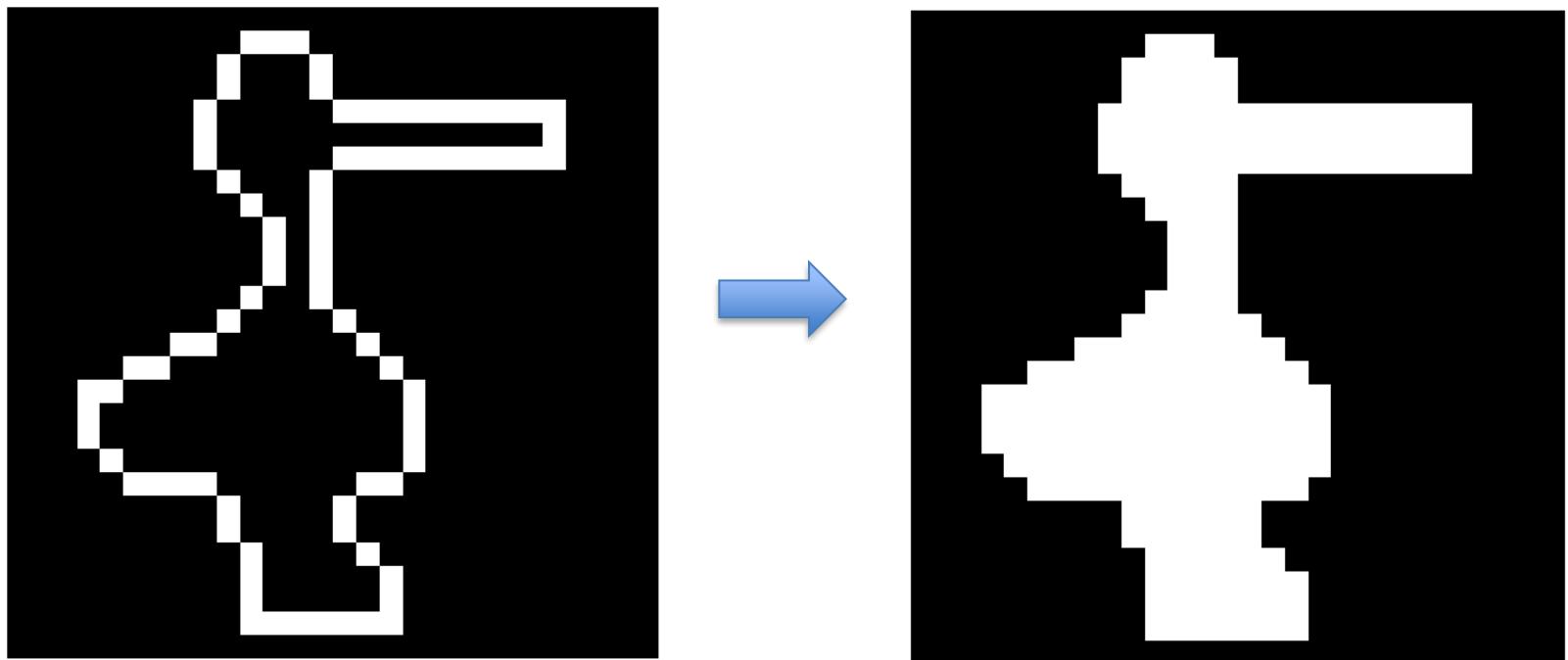
# Ejemplos: Erosión



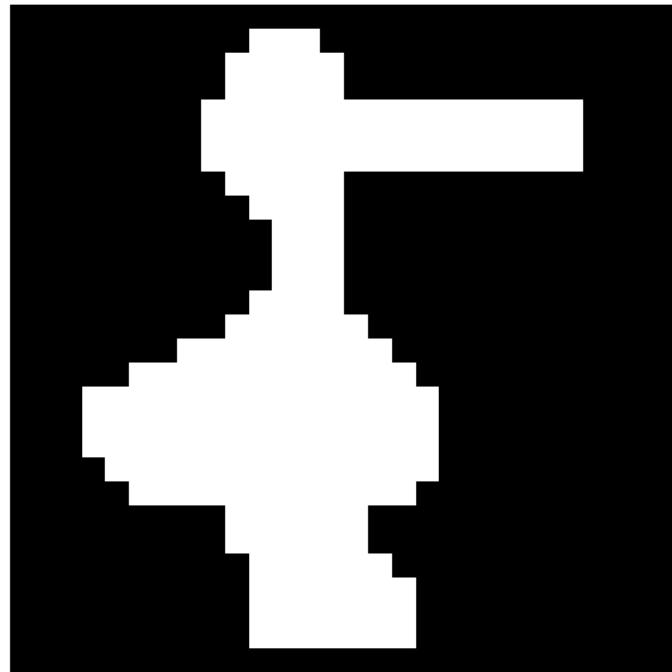
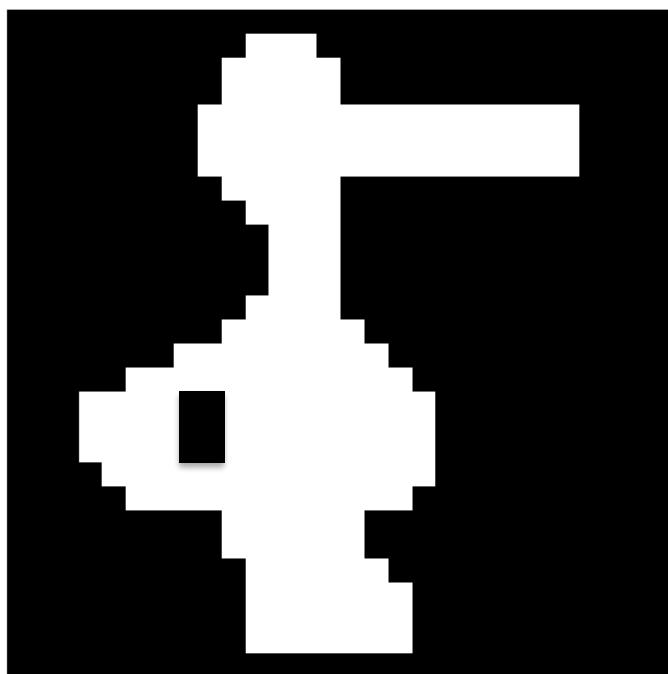
# Ejemplos: Perímetro



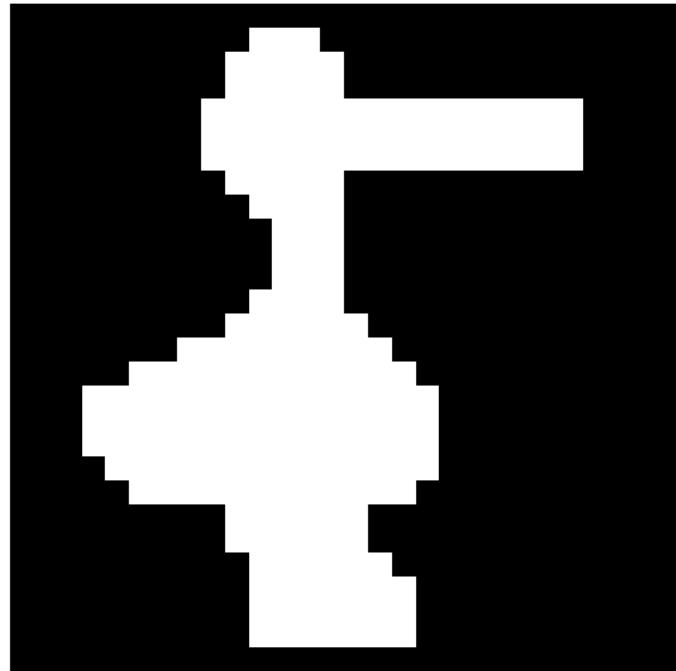
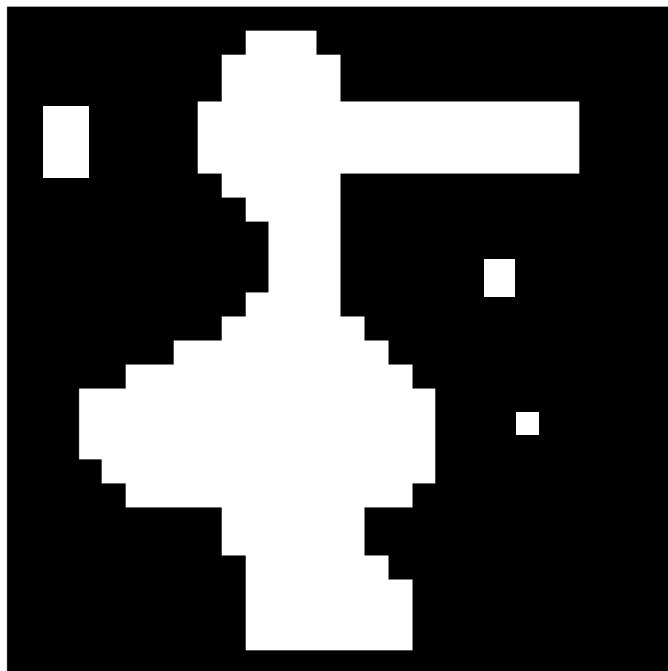
## Ejemplos: Relleno



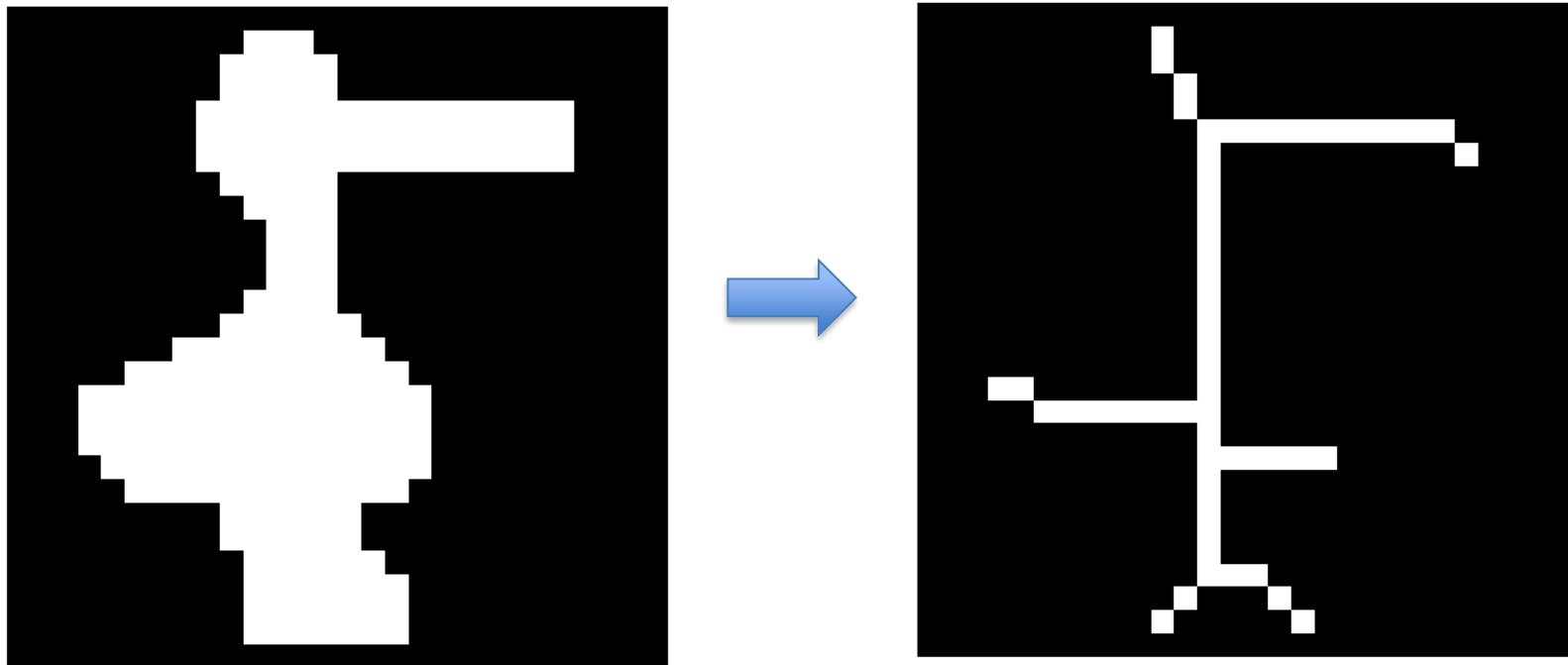
# Ejemplos: Relleno



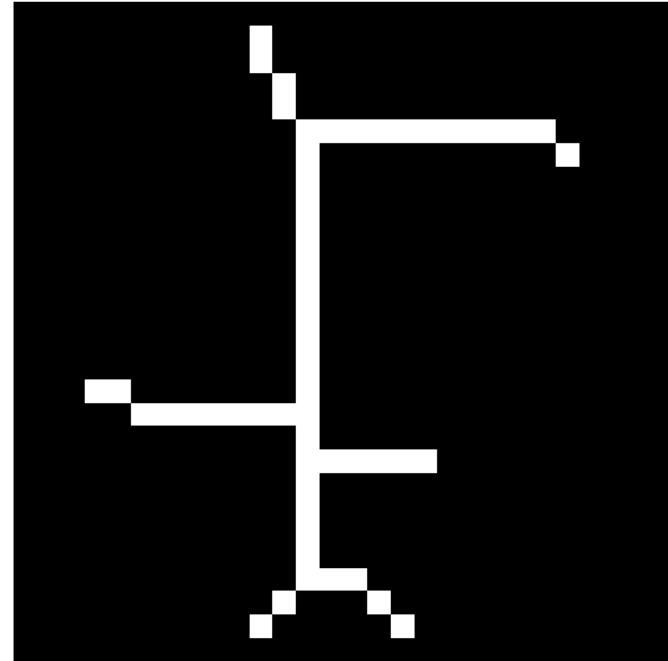
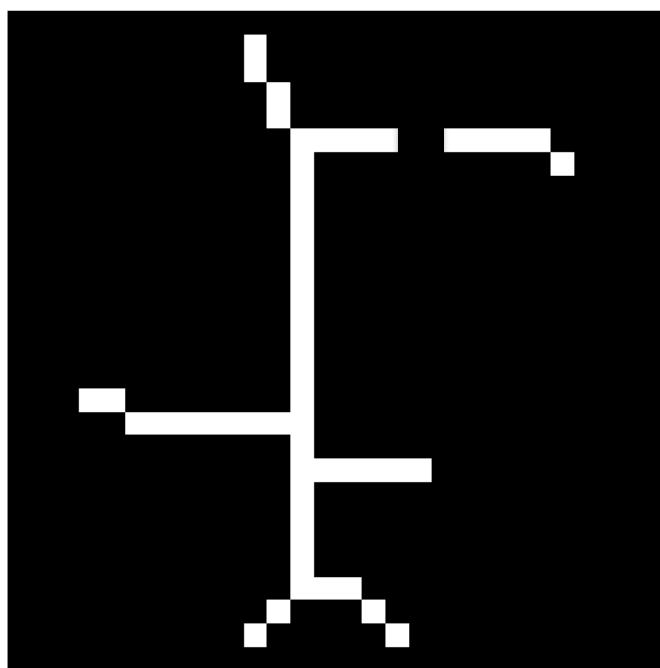
# Ejemplos: Limpieza



# Ejemplos: Esqueletización



# Ejemplos: Unión

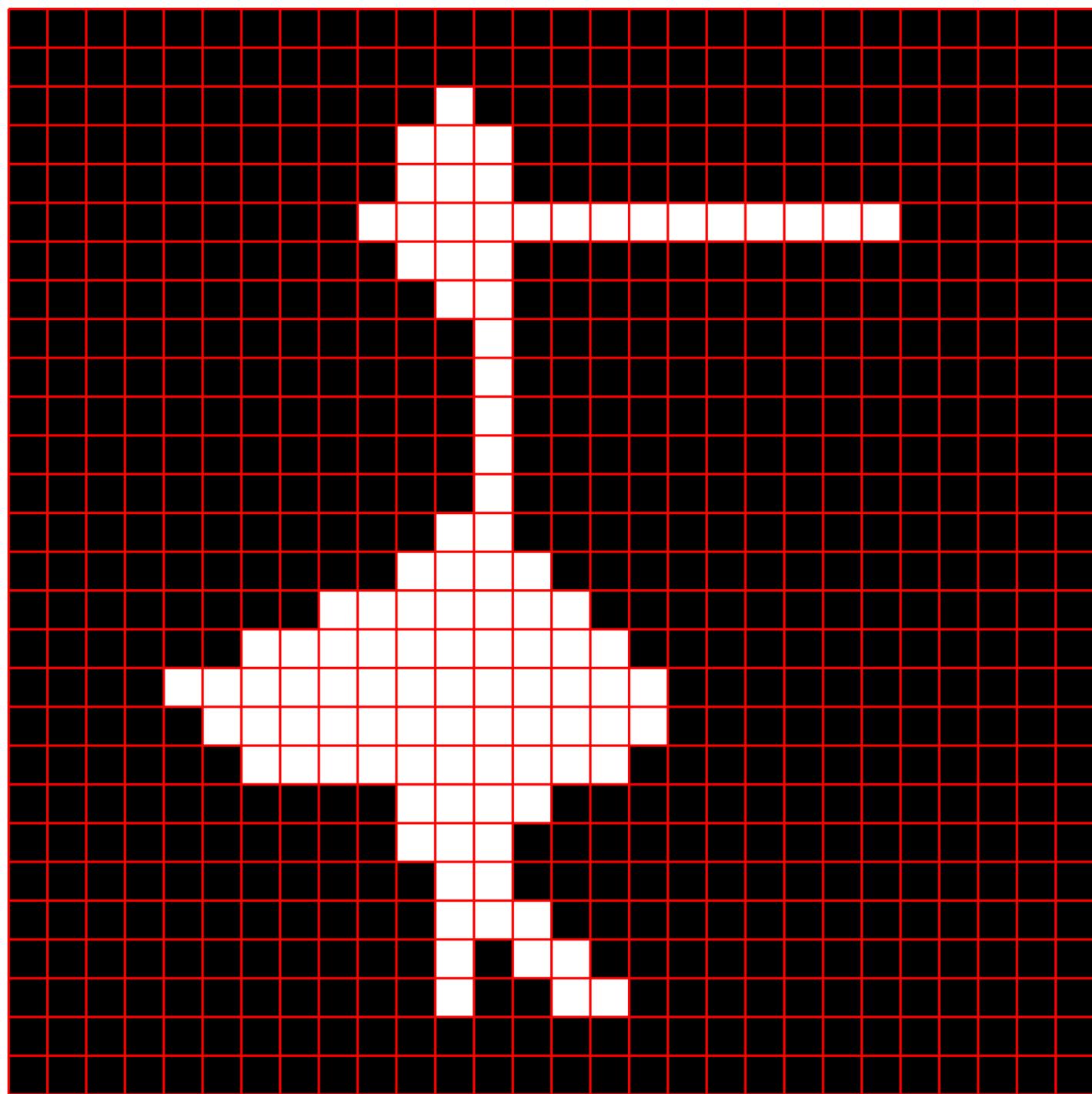


# Dilatación

$$Y = X \odot S$$

|      |      |  
Output   Input   Estructura

Ejemplo  
Imagen  
binaria  
 $28 \times 28$

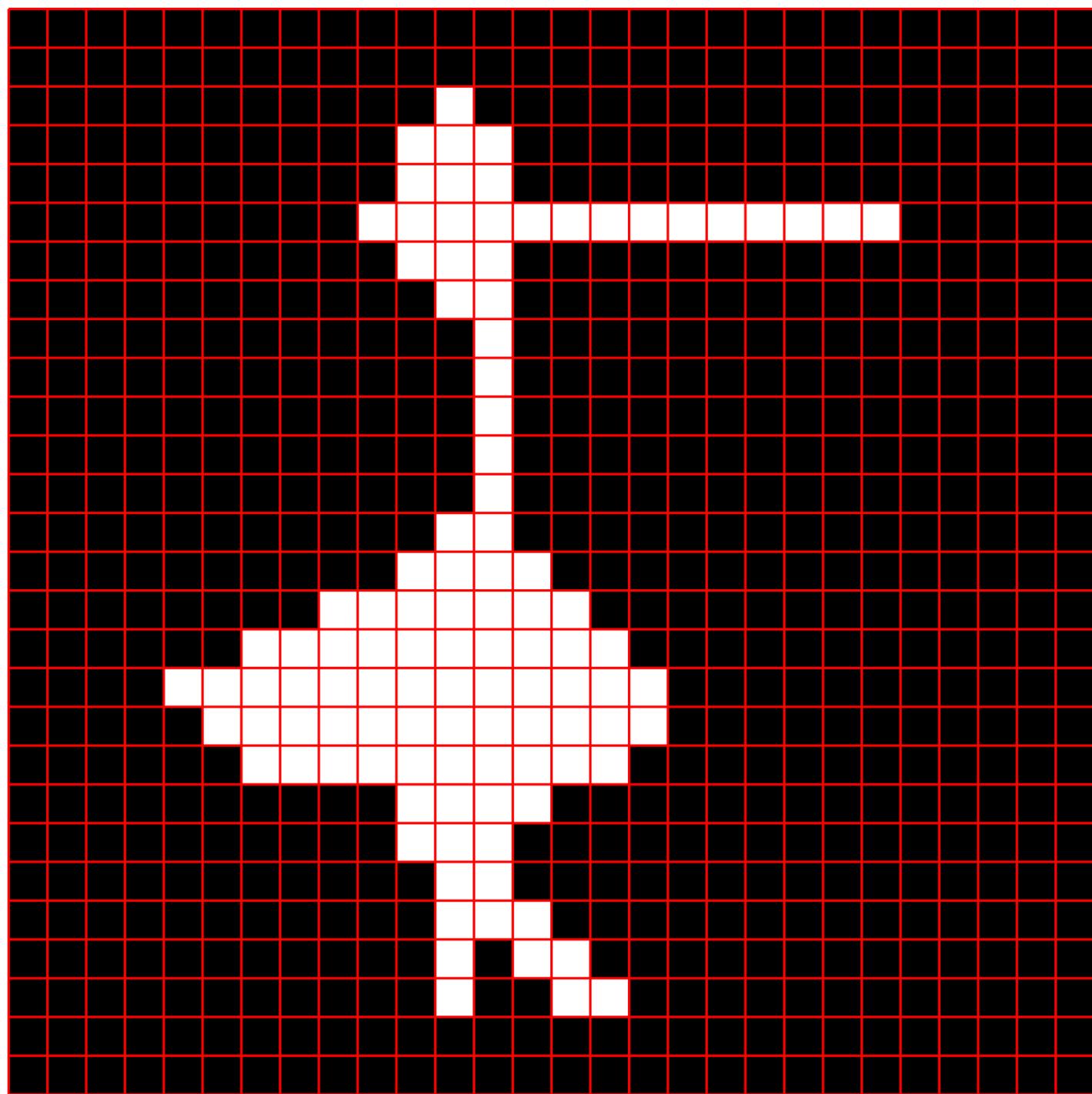


# Ejemplo Imagen binaria 28 x28

Ejemplo  
Imagen  
binaria  
28 x28

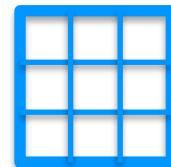
Ejemplo  
Imagen  
binaria  
28 x28

Ejemplo  
Imagen  
binaria  
 $28 \times 28$



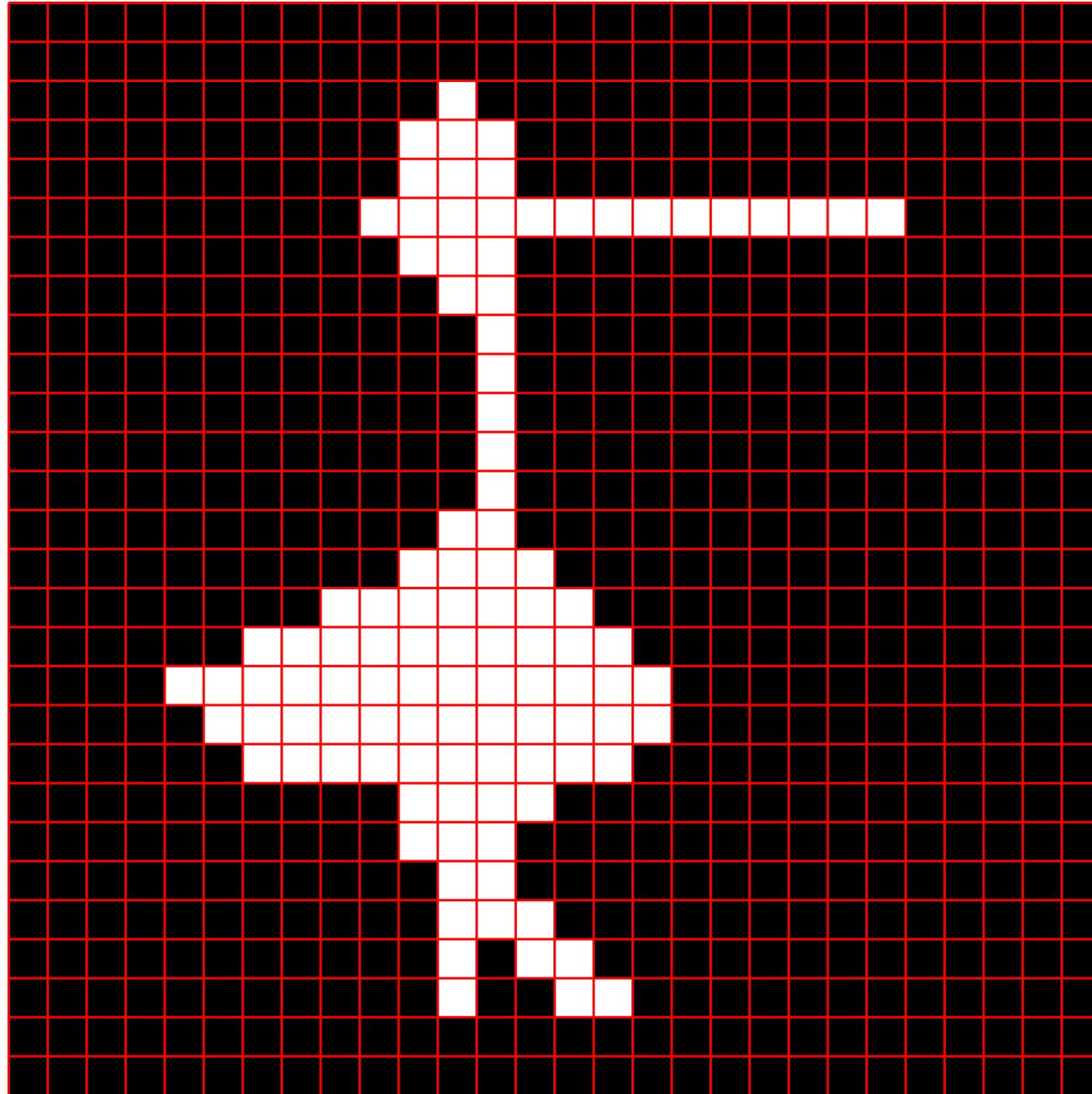
# Ejemplo Imagen binaria $28 \times 28$

Estructura

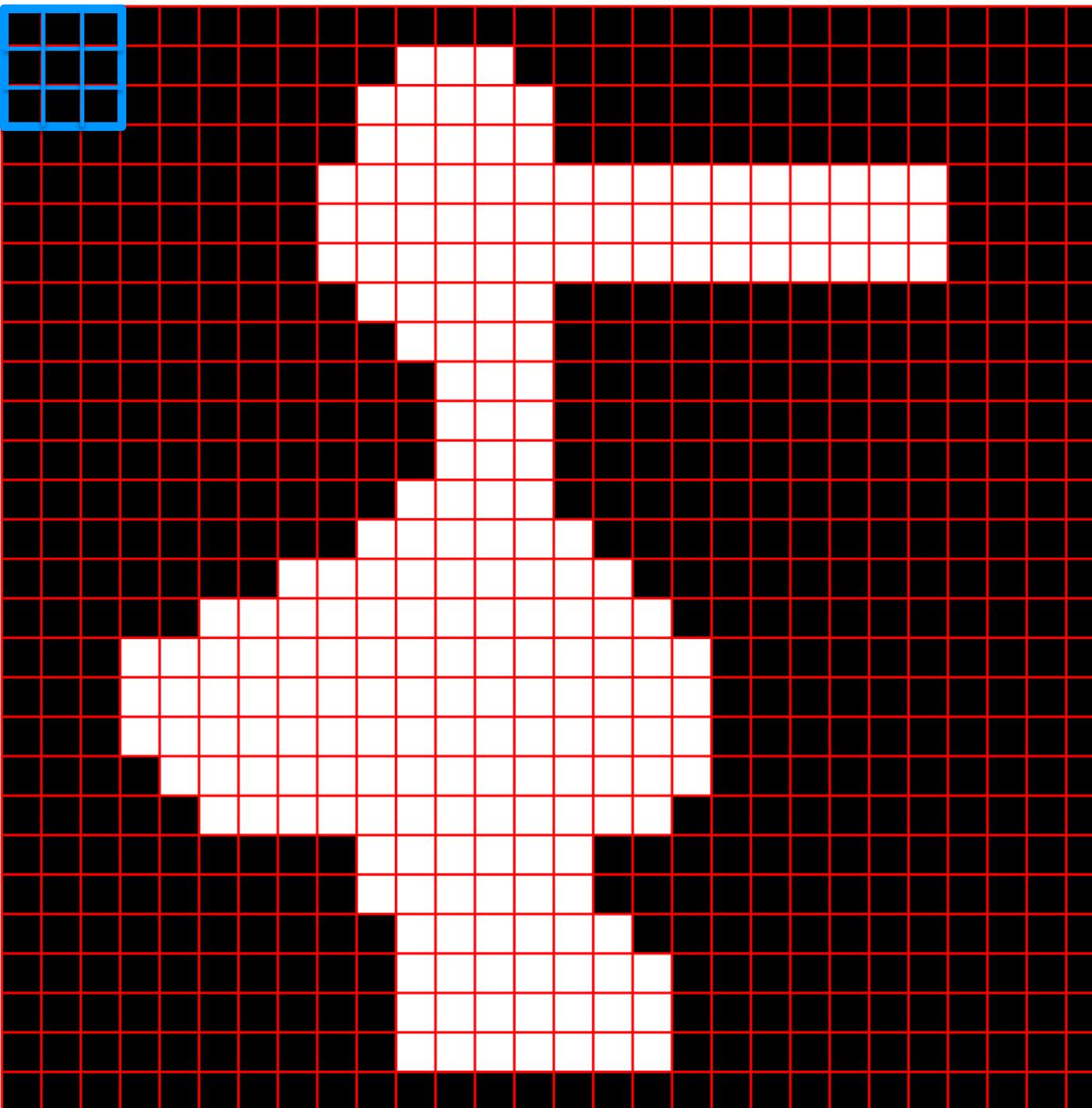


$3 \times 3$

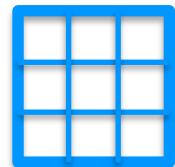
Ventana deslizante:  
la salida es cero,  
sólo si la estructura  
cubre sólo ceros, de  
lo contrario es 1.



# Ejemplo Imagen binaria $28 \times 28$



Estructura

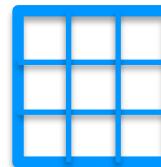


$3 \times 3$

Ventana deslizante:  
la salida es cero,  
sólo si la estructura  
cubre sólo ceros, de  
lo contrario es 1.

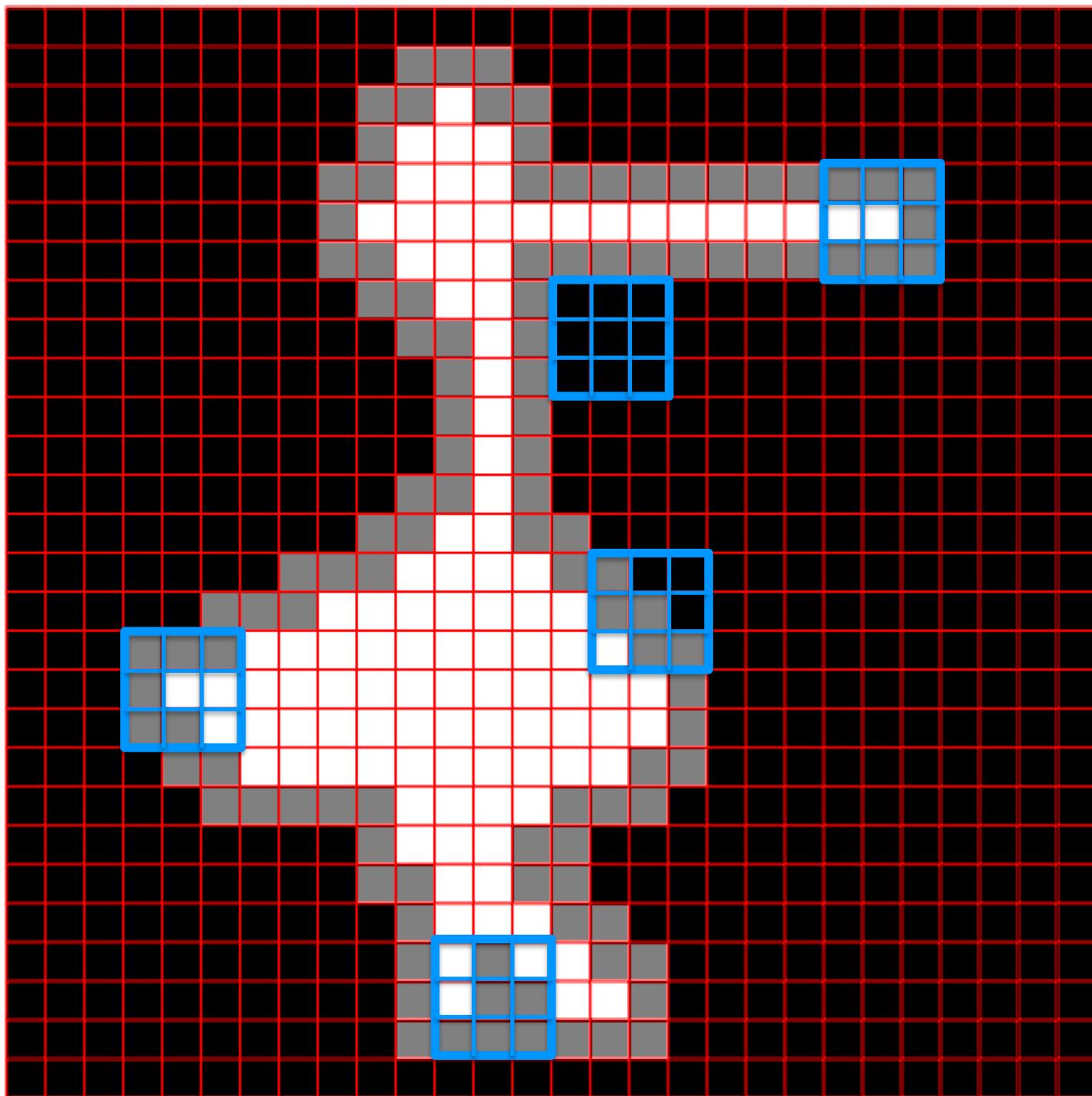
# Ejemplo Imagen binaria $28 \times 28$

Estructura



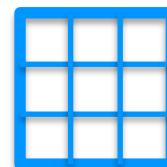
$3 \times 3$

Ventana deslizante:  
la salida es cero,  
sólo si la estructura  
cubre sólo ceros, de  
lo contrario es 1.



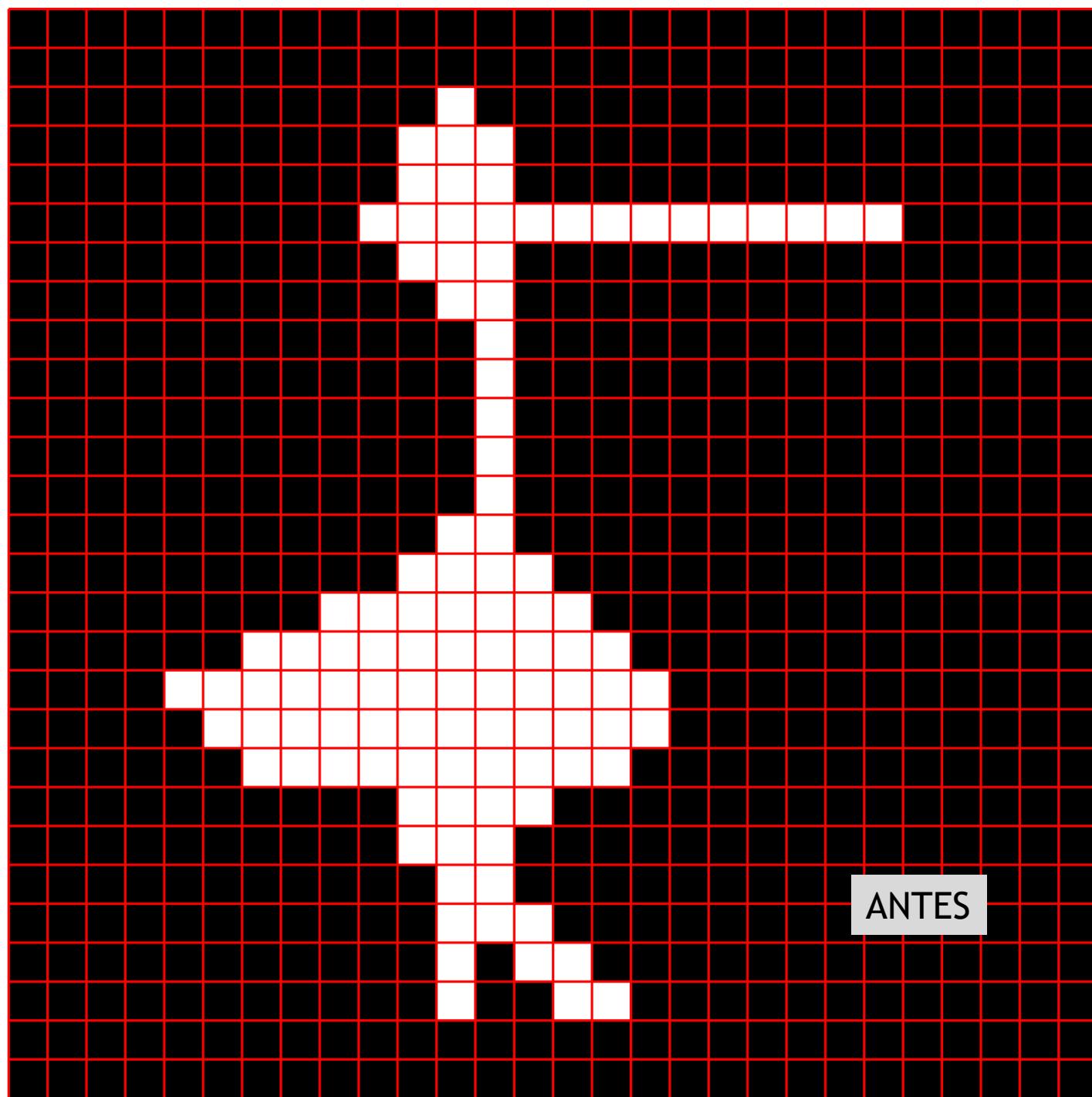
# Ejemplo Imagen binaria $28 \times 28$

Estructura



$3 \times 3$

Ventana deslizante:  
la salida es cero,  
sólo si la estructura  
cubre sólo ceros, de  
lo contrario es 1.



# Ejemplo Imagen binaria $28 \times 28$

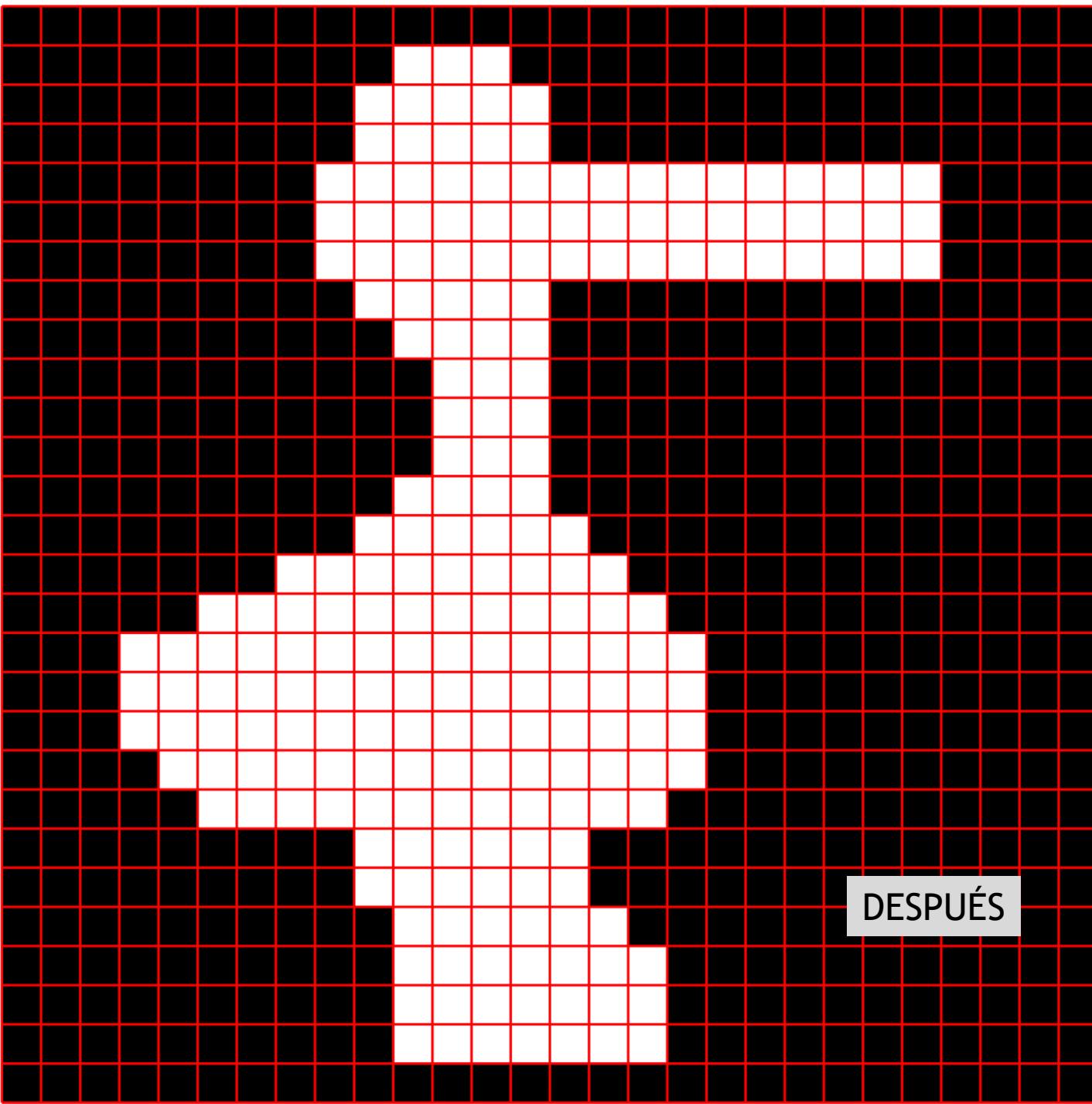
Estructura



$3 \times 3$

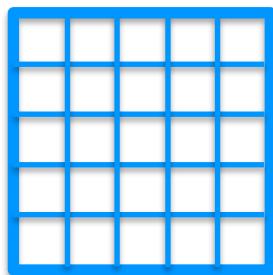
Ventana deslizante:  
la salida es cero,  
sólo si la estructura  
cubre sólo ceros, de  
lo contrario es 1.

DESPUÉS

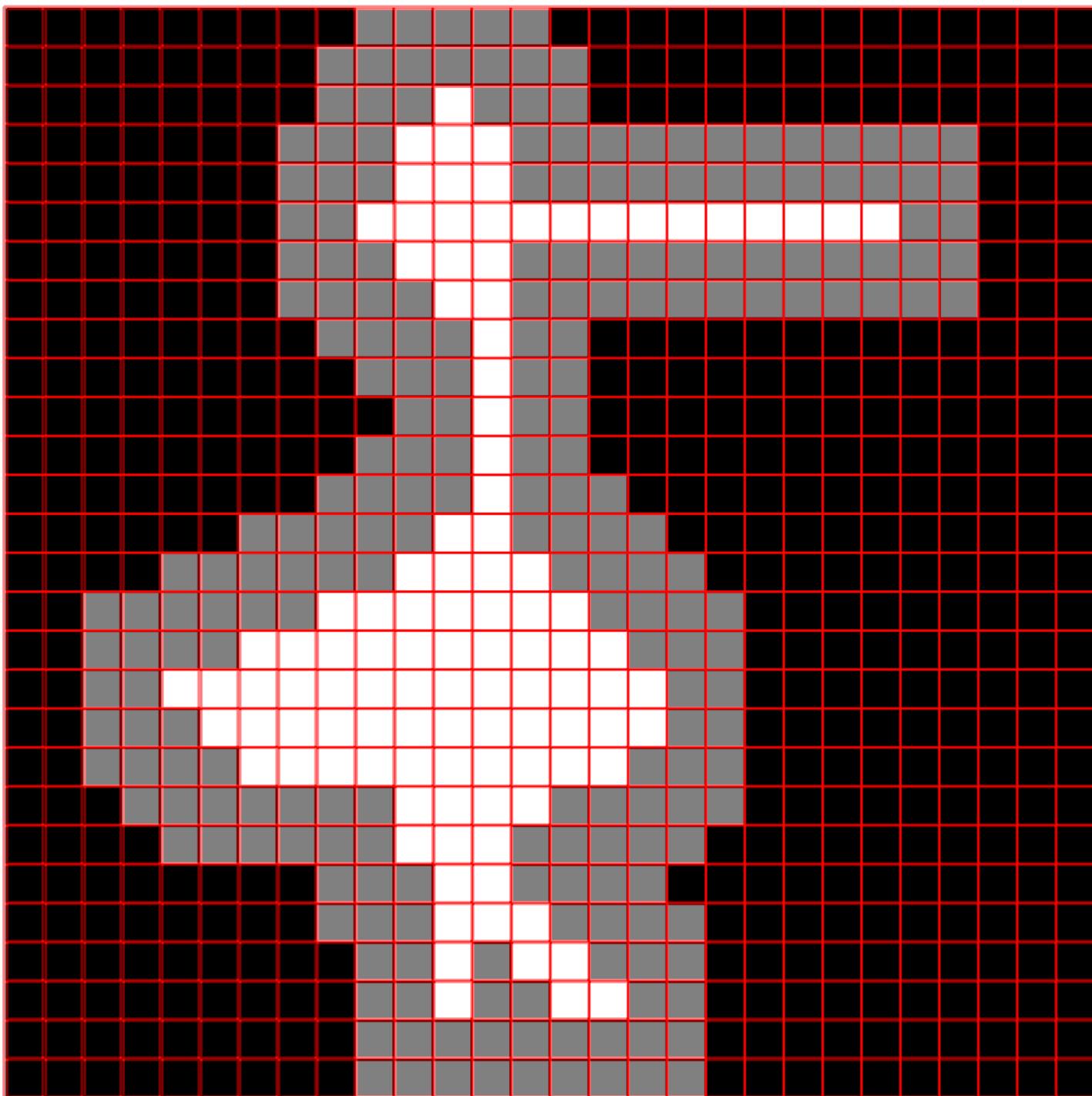


Ejemplo  
Imagen  
binaria  
 $28 \times 28$

Estructura

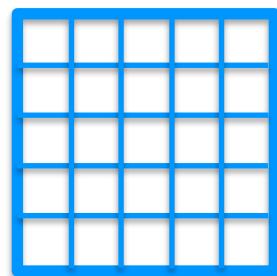


5x5

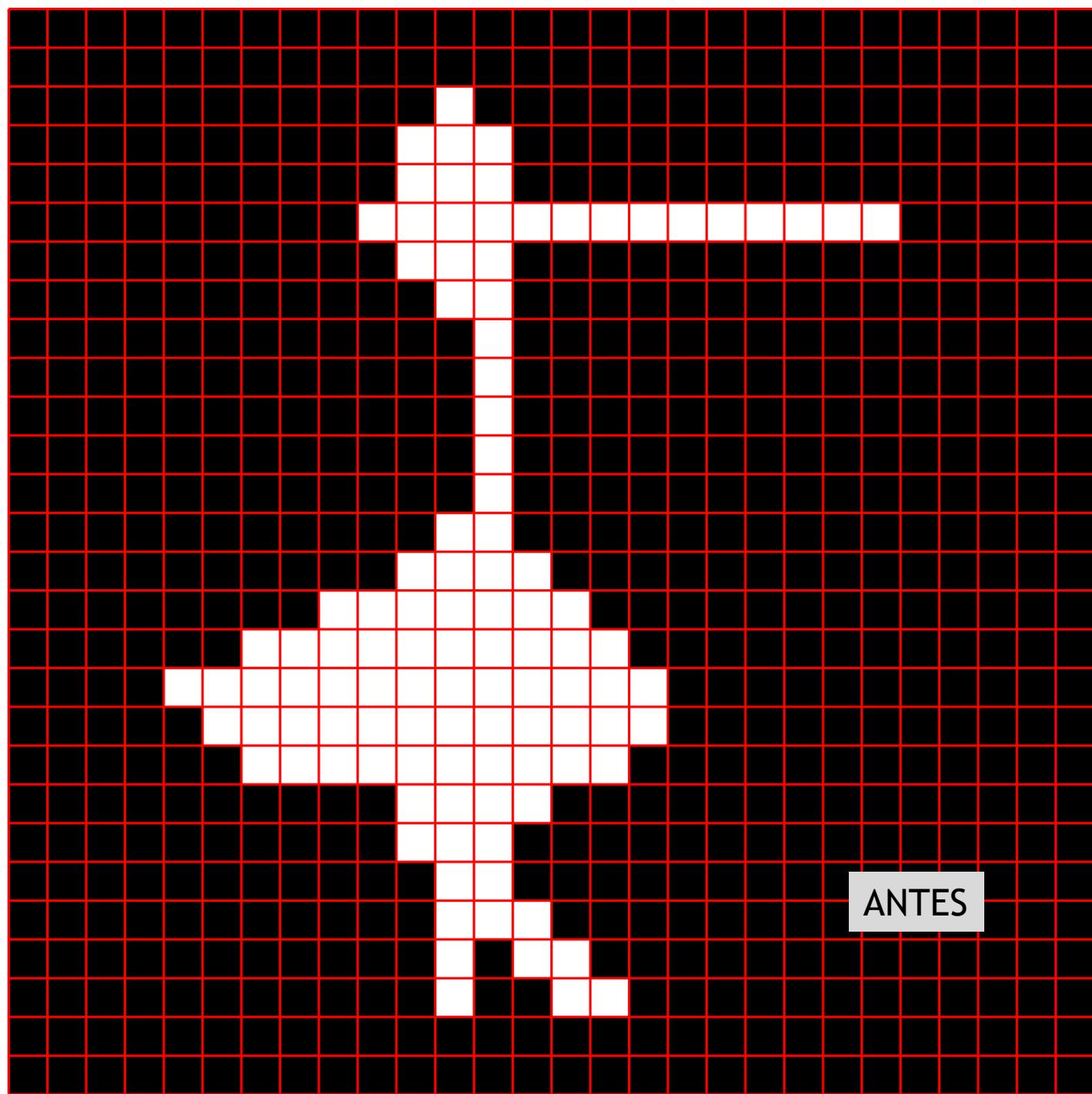


Ejemplo  
Imagen  
binaria  
 $28 \times 28$

Estructura

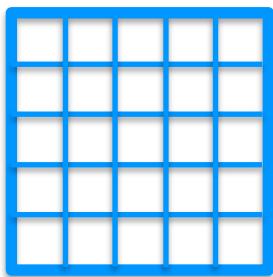


$5 \times 5$



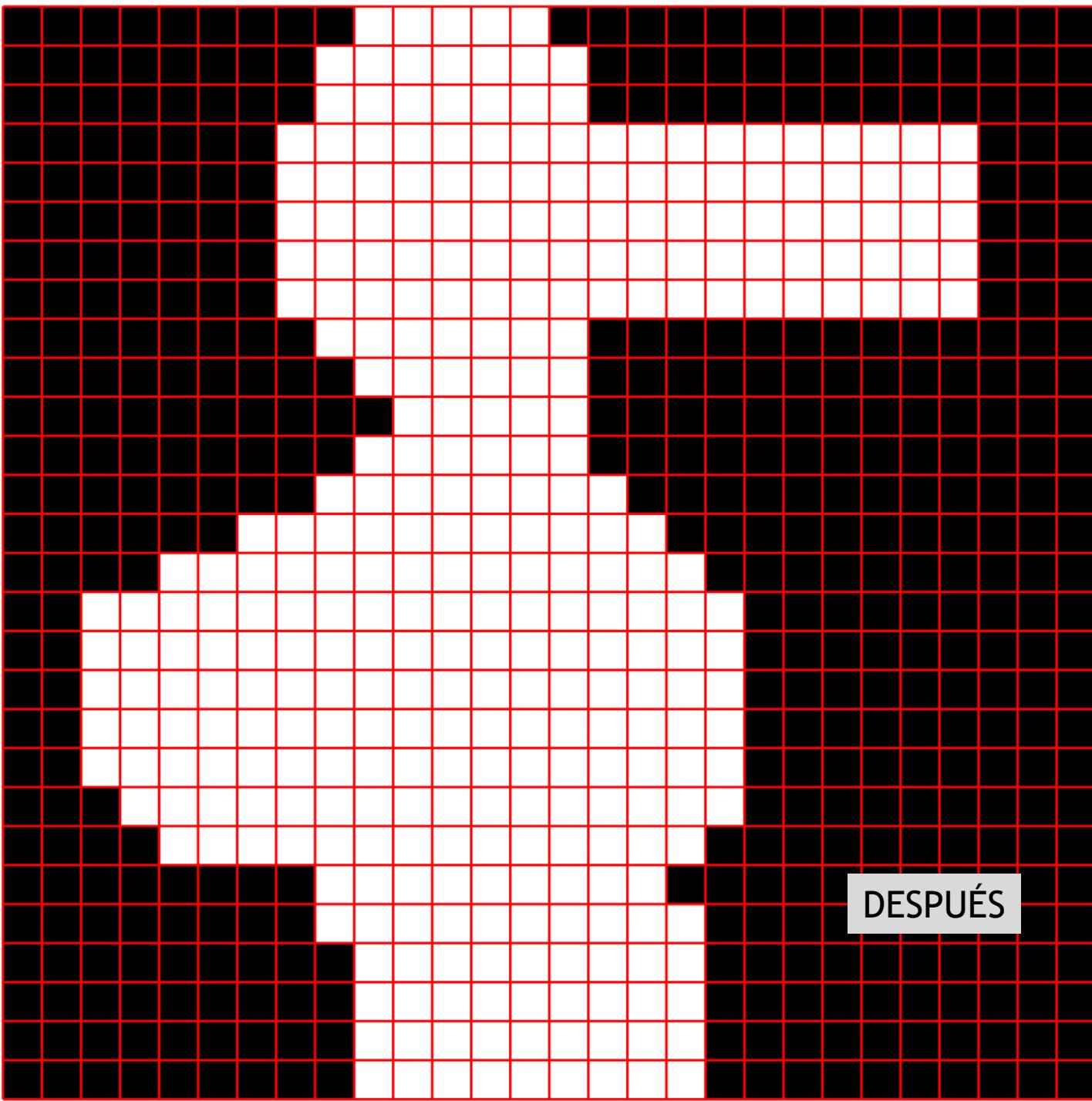
Ejemplo  
Imagen  
binaria  
 $28 \times 28$

Estructura



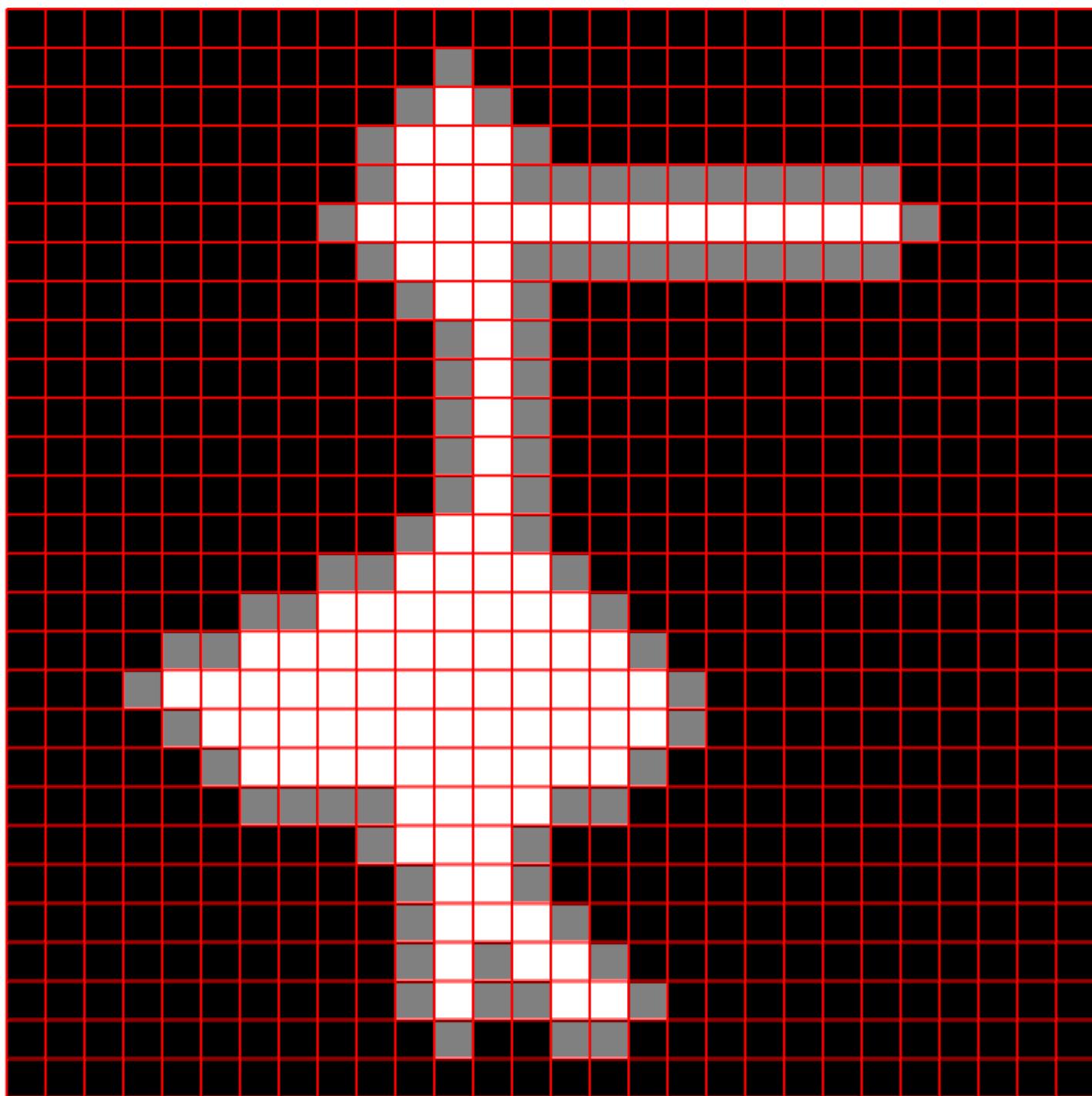
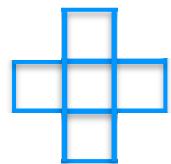
$5 \times 5$

DESPUÉS



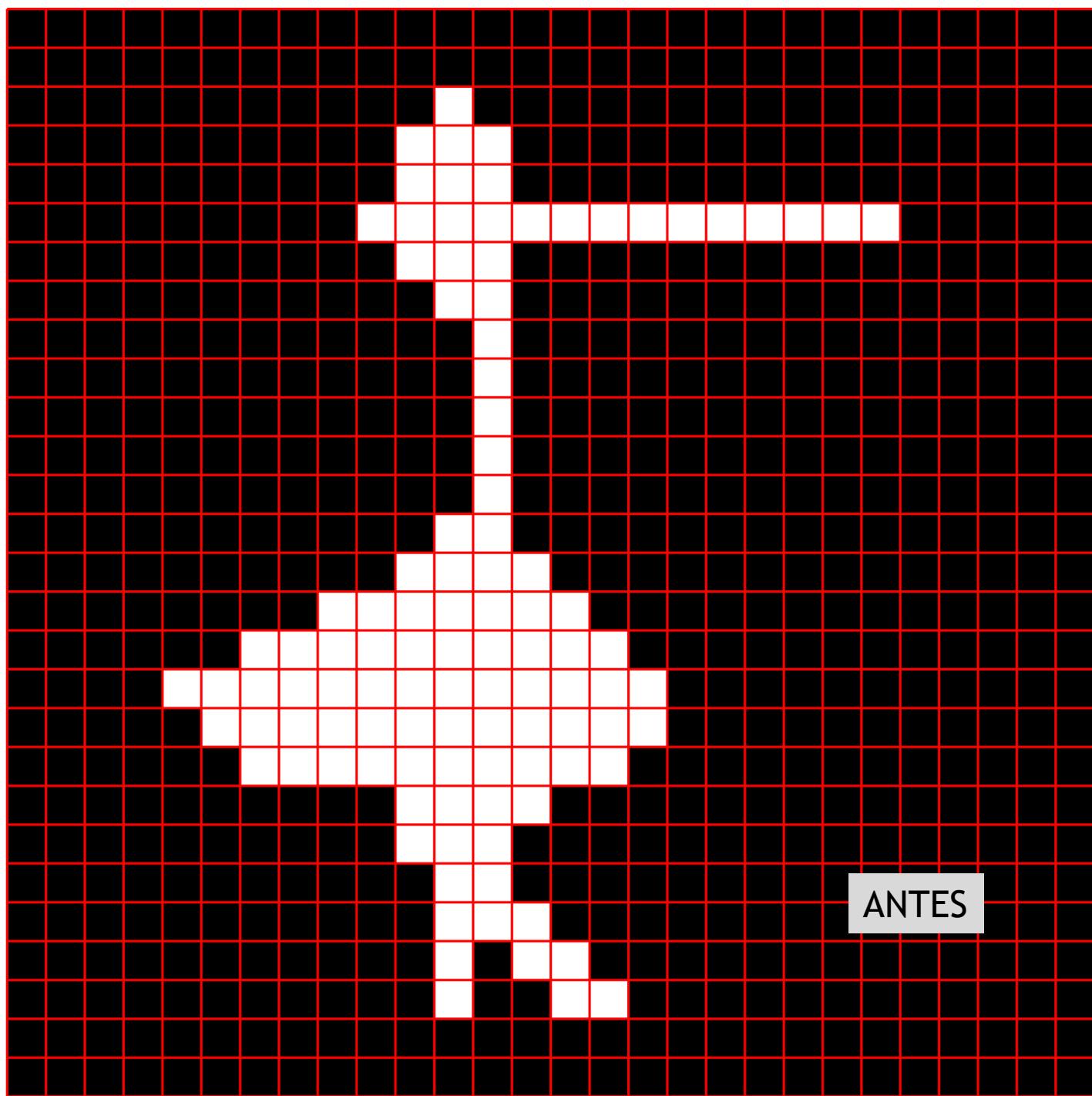
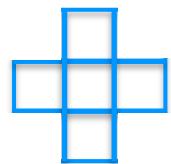
Ejemplo  
Imagen  
binaria  
 $28 \times 28$

Estructura



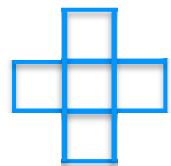
Ejemplo  
Imagen  
binaria  
 $28 \times 28$

Estructura

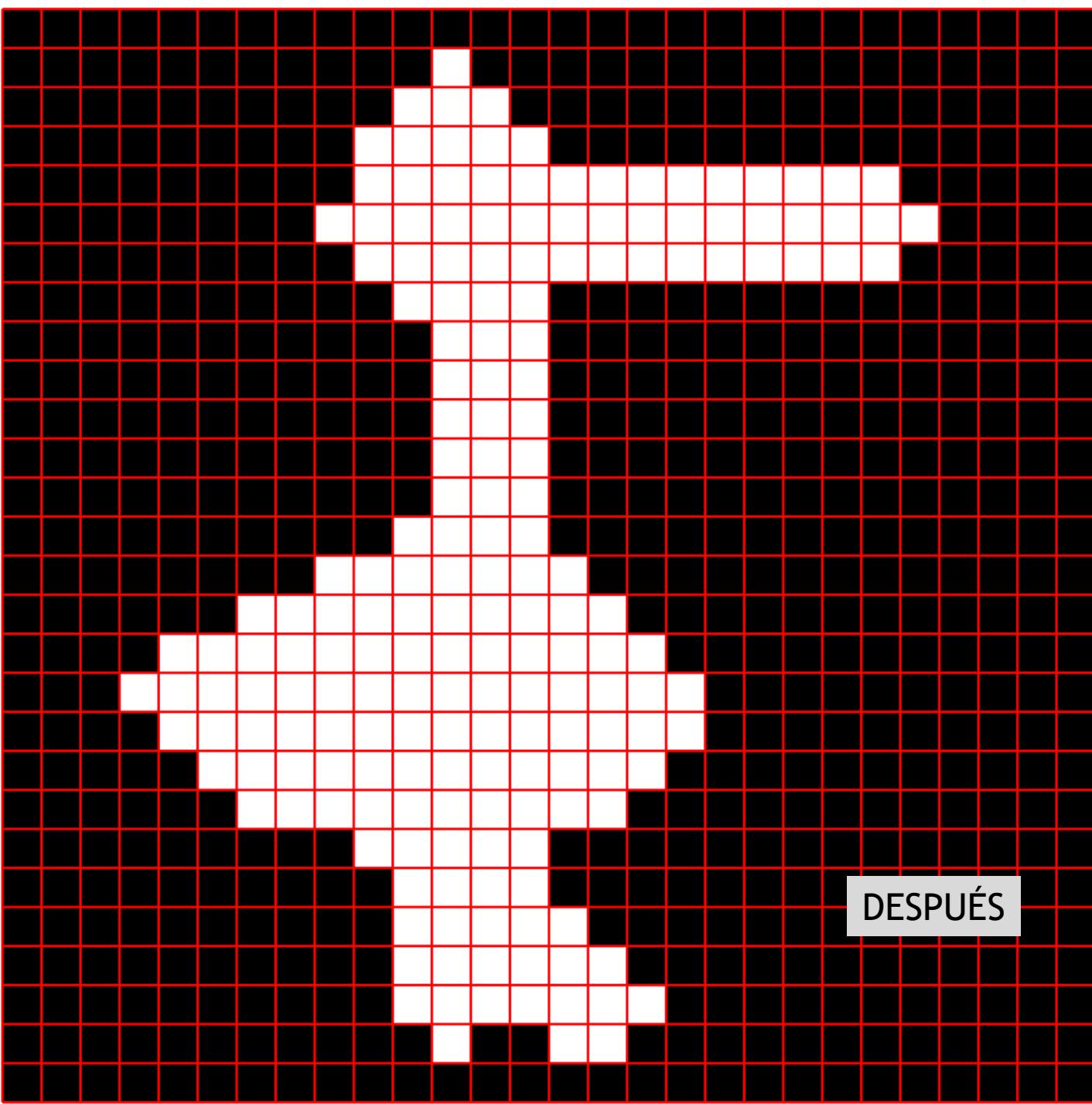


Ejemplo  
Imagen  
binaria  
 $28 \times 28$

Estructura



DESPUÉS



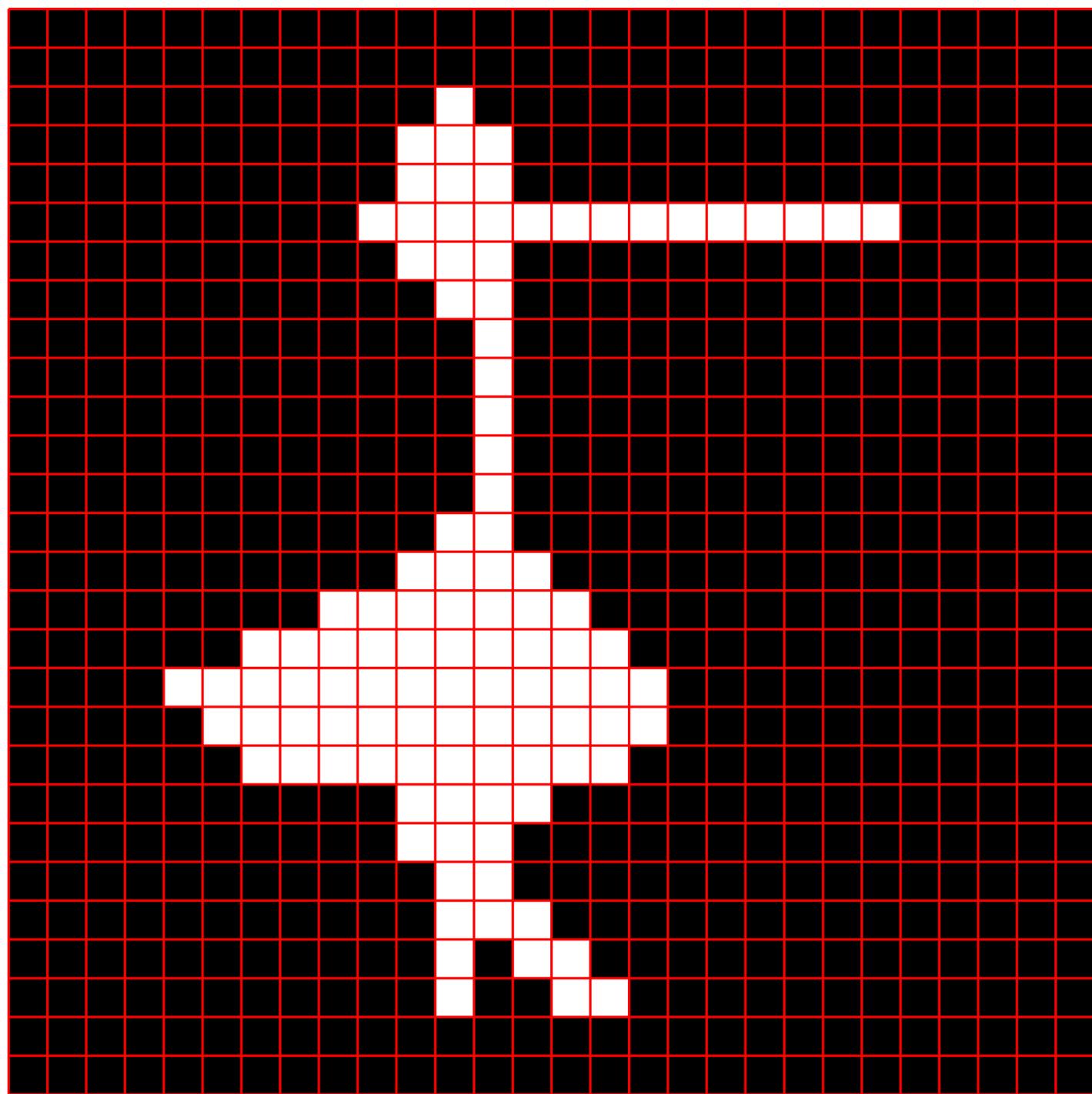
# Erosión

$$Y = X \ominus S$$

|            |            |  
Output      Input      Estructura

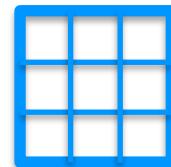
# Ejemplo Imagen binaria 28 x28

Ejemplo  
Imagen  
binaria  
 $28 \times 28$



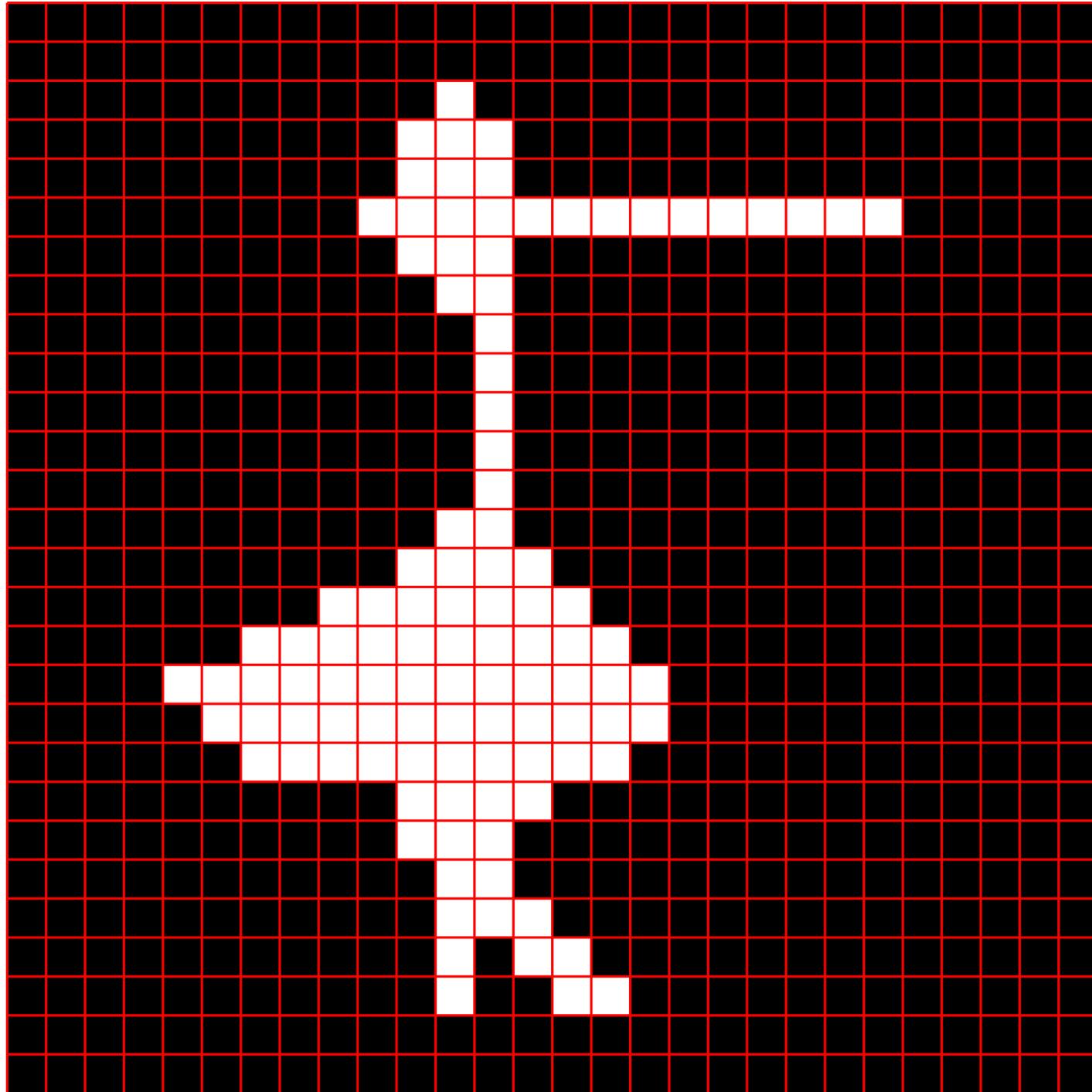
# Ejemplo Imagen binaria $28 \times 28$

Estructura



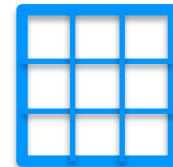
$3 \times 3$

Ventana deslizante:  
la salida es uno, sólo  
si la estructura  
contiene sólo unos,  
de lo contrario es 0.



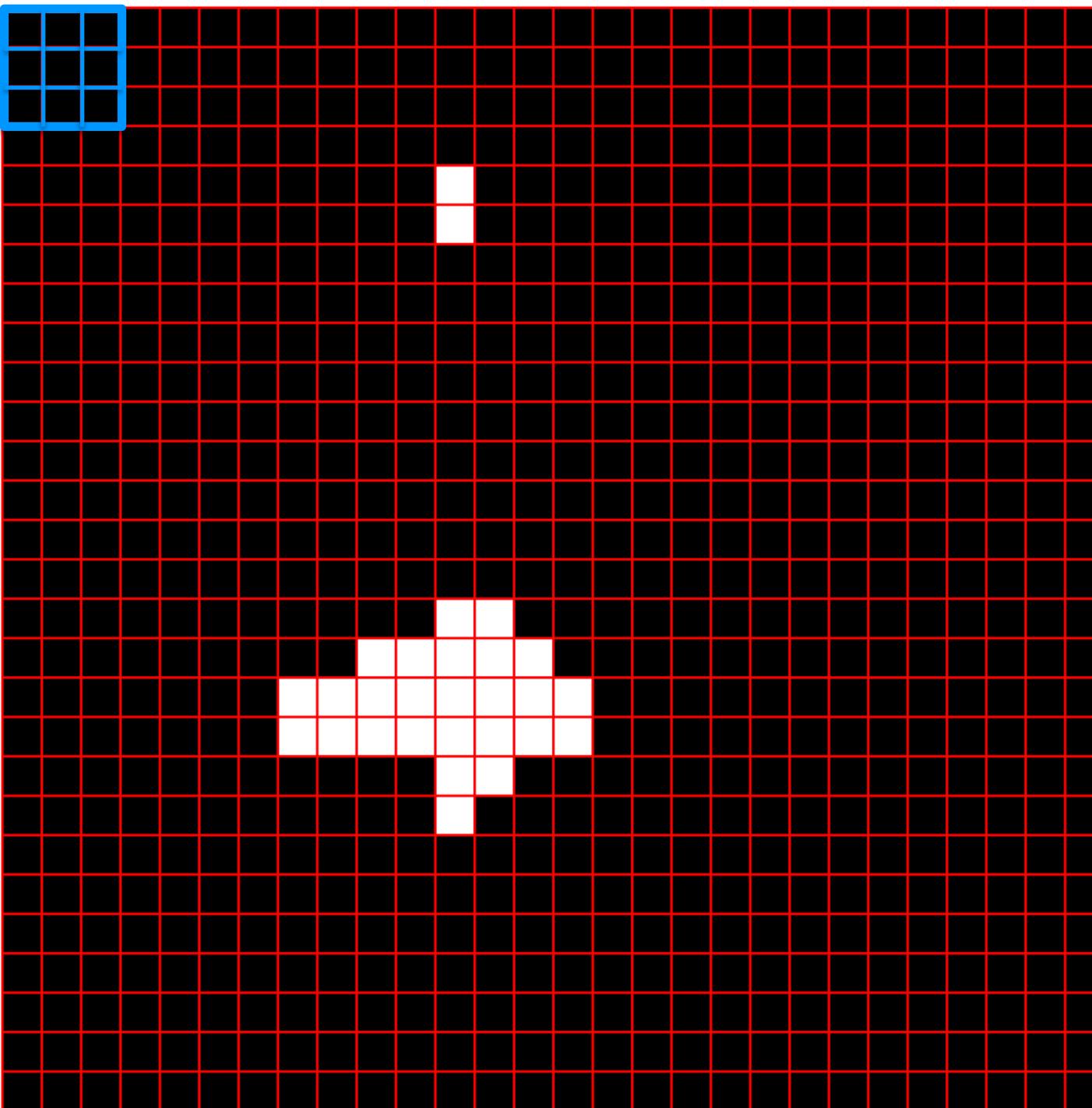
# Ejemplo Imagen binaria $28 \times 28$

Estructura



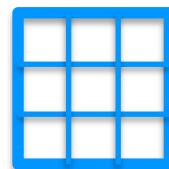
$3 \times 3$

Ventana deslizante:  
la salida es uno, sólo  
si la estructura  
cubre sólo unos, de  
lo contrario es 0.



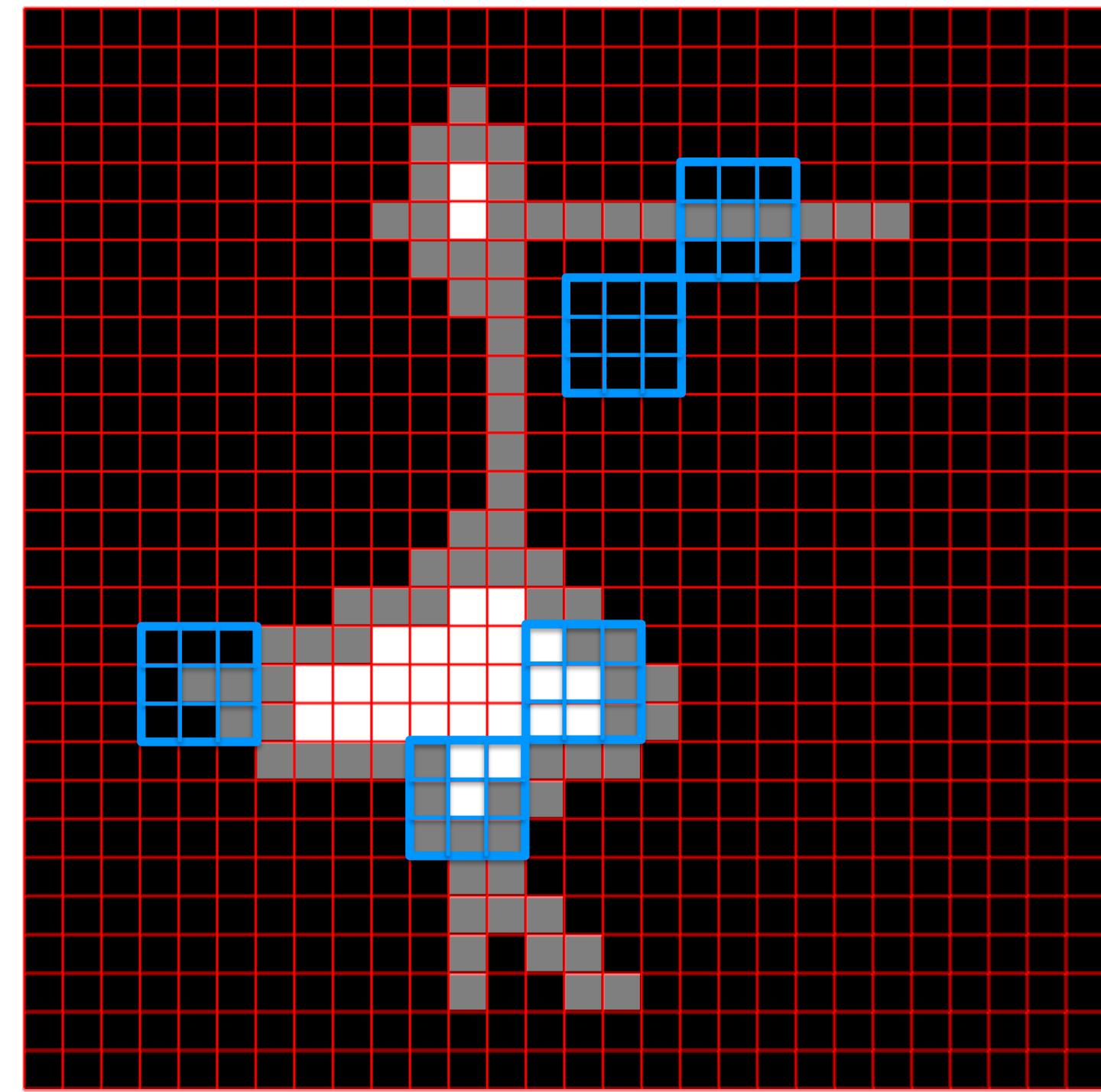
# Ejemplo Imagen binaria $28 \times 28$

Estructura



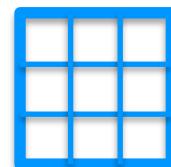
$3 \times 3$

Ventana deslizante:  
la salida es uno, sólo  
si la estructura  
cubre sólo unos, de  
lo contrario es 0.



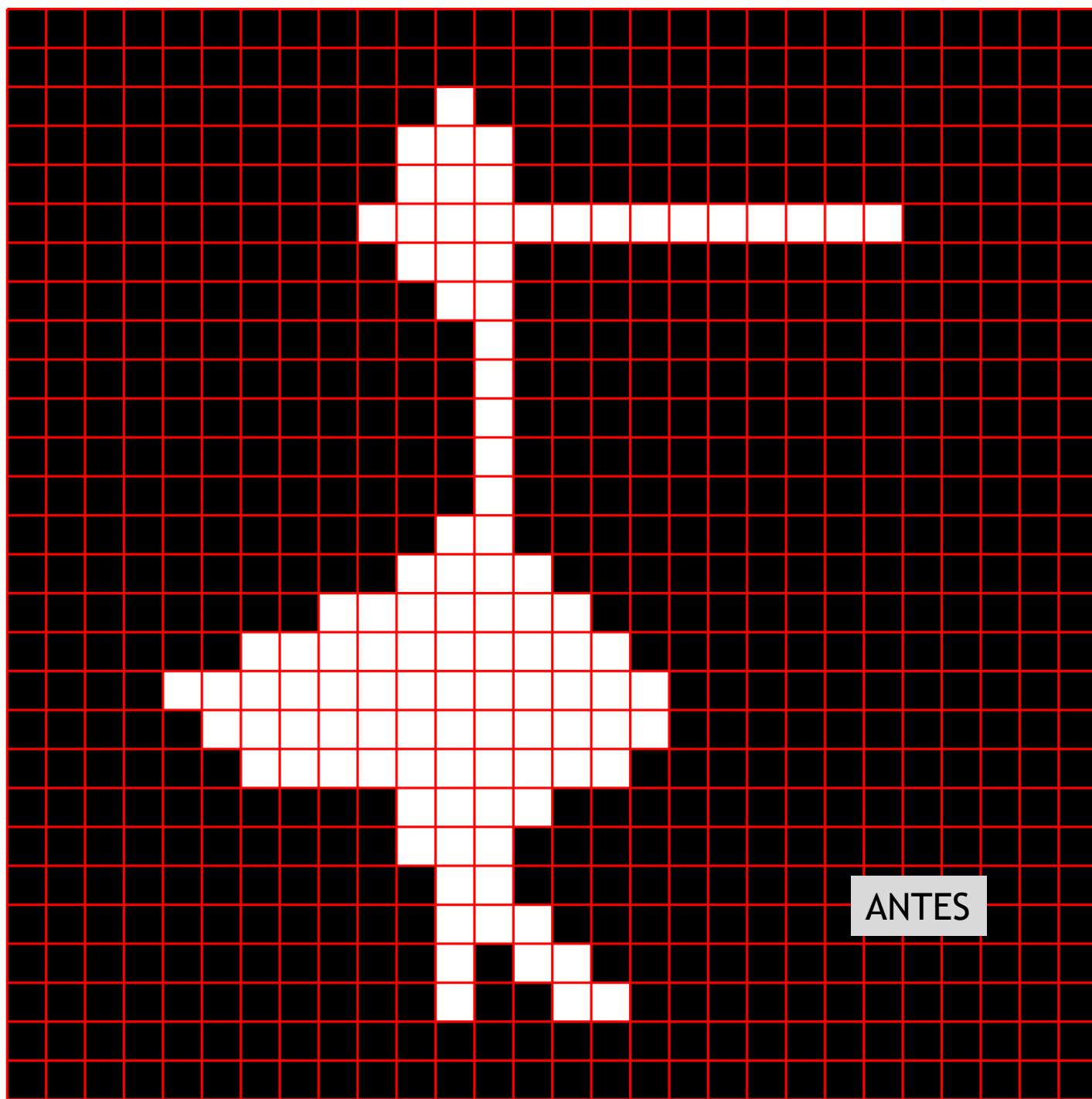
# Ejemplo Imagen binaria $28 \times 28$

Estructura



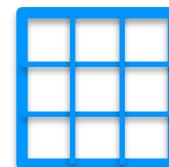
$3 \times 3$

Ventana deslizante:  
la salida es uno, sólo  
si la estructura  
cubre sólo unos, de  
lo contrario es 0.



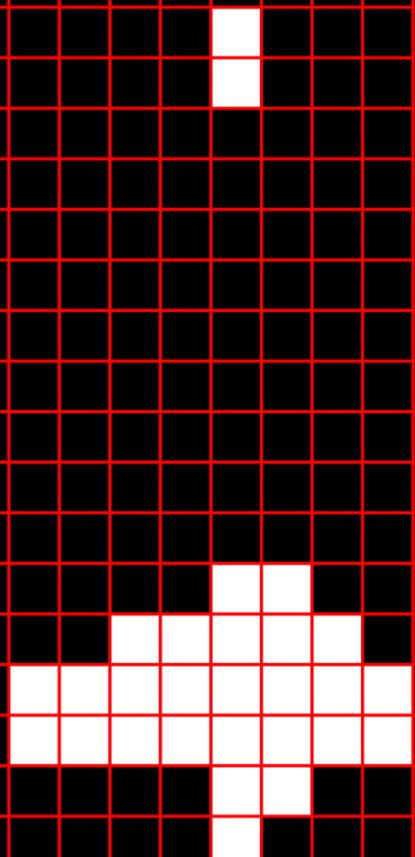
# Ejemplo Imagen binaria $28 \times 28$

Estructura



$3 \times 3$

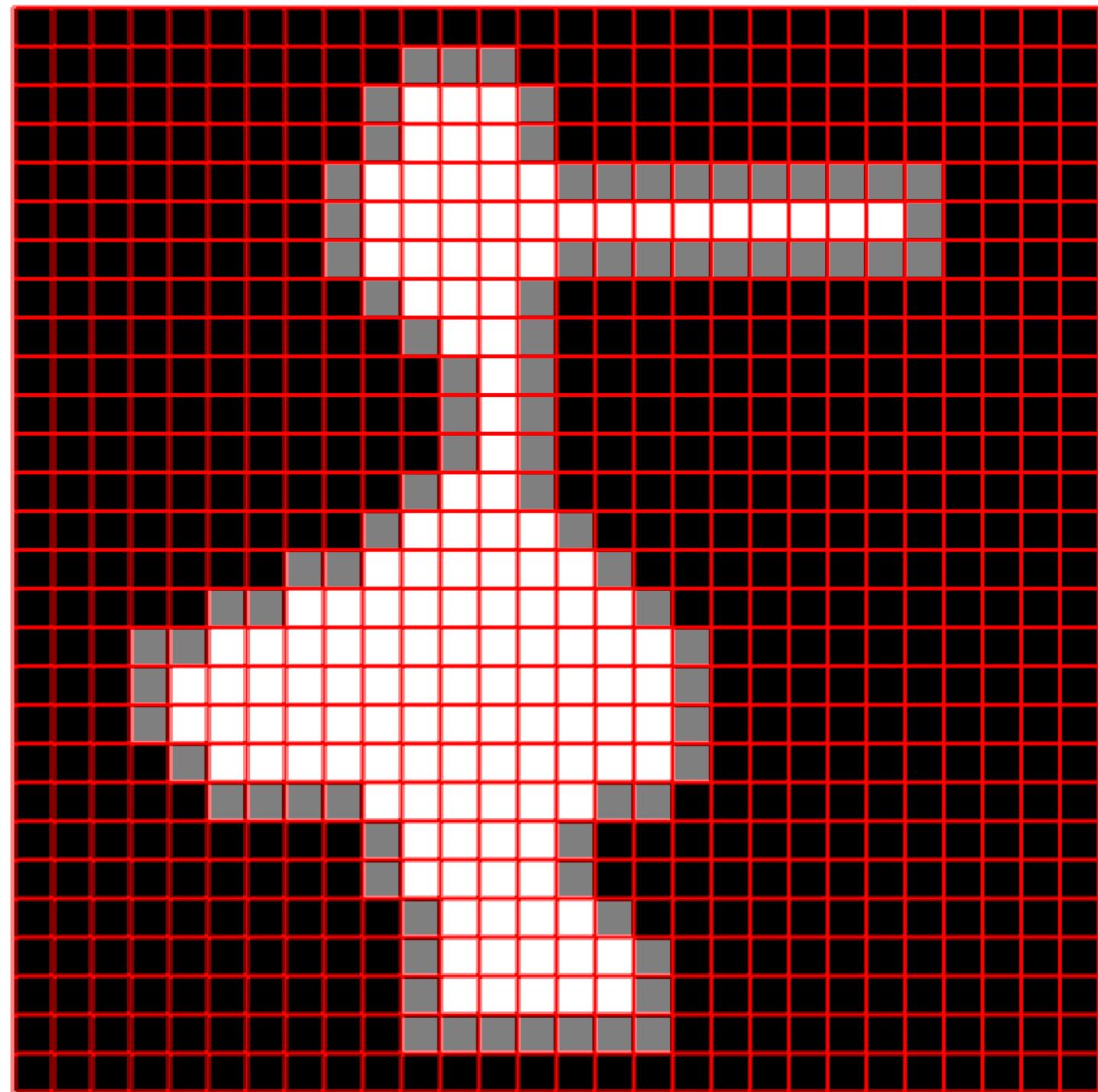
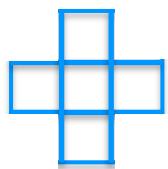
Ventana deslizante:  
la salida es uno, sólo  
si la estructura  
cubre sólo unos, de  
lo contrario es 0.



DESPUÉS

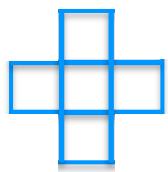
Ejemplo  
Imagen  
binaria  
 $28 \times 28$

Erosión con

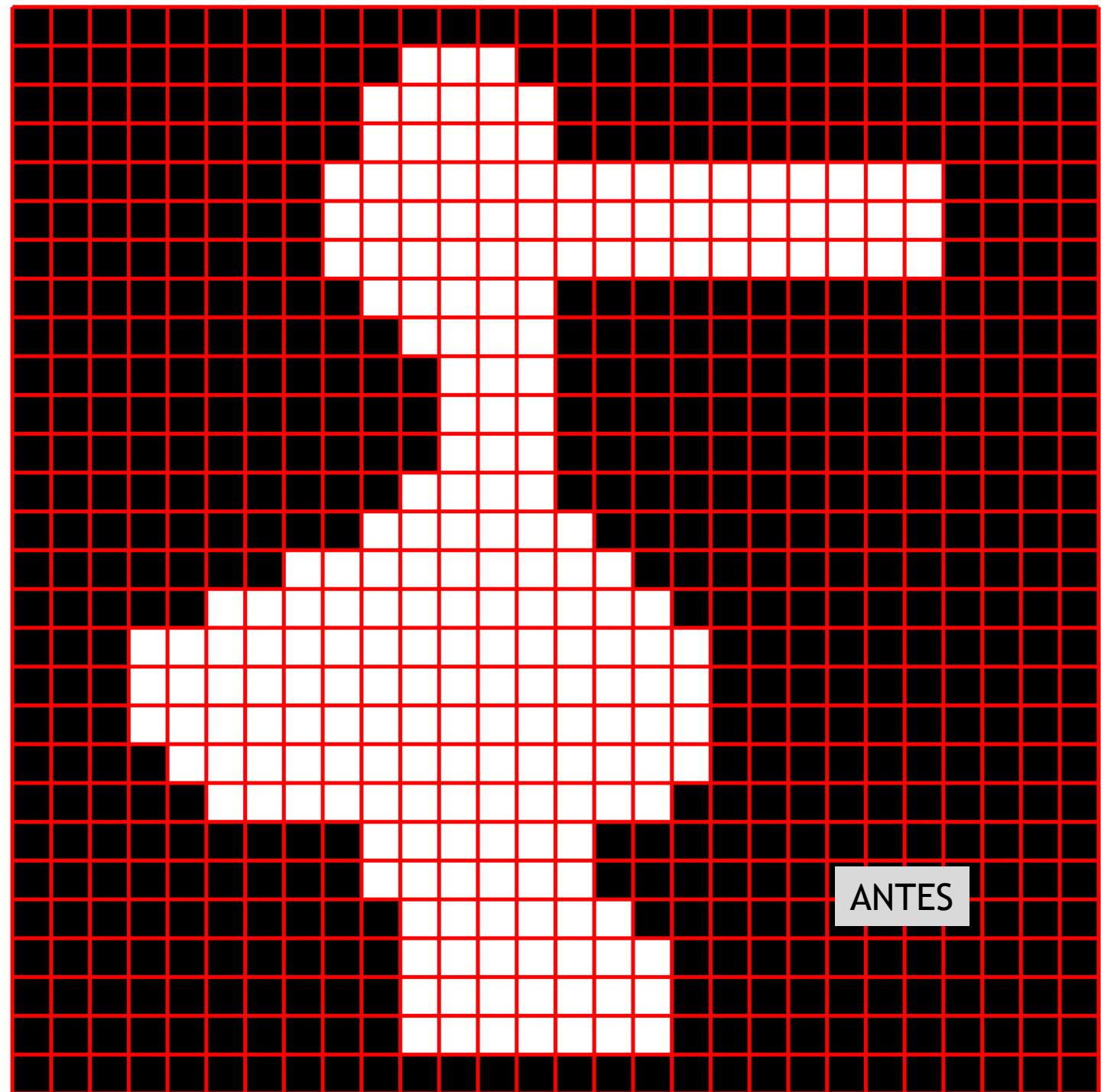


# Ejemplo Imagen binaria $28 \times 28$

Erosión con

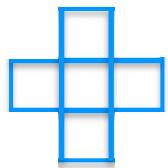


ANTES

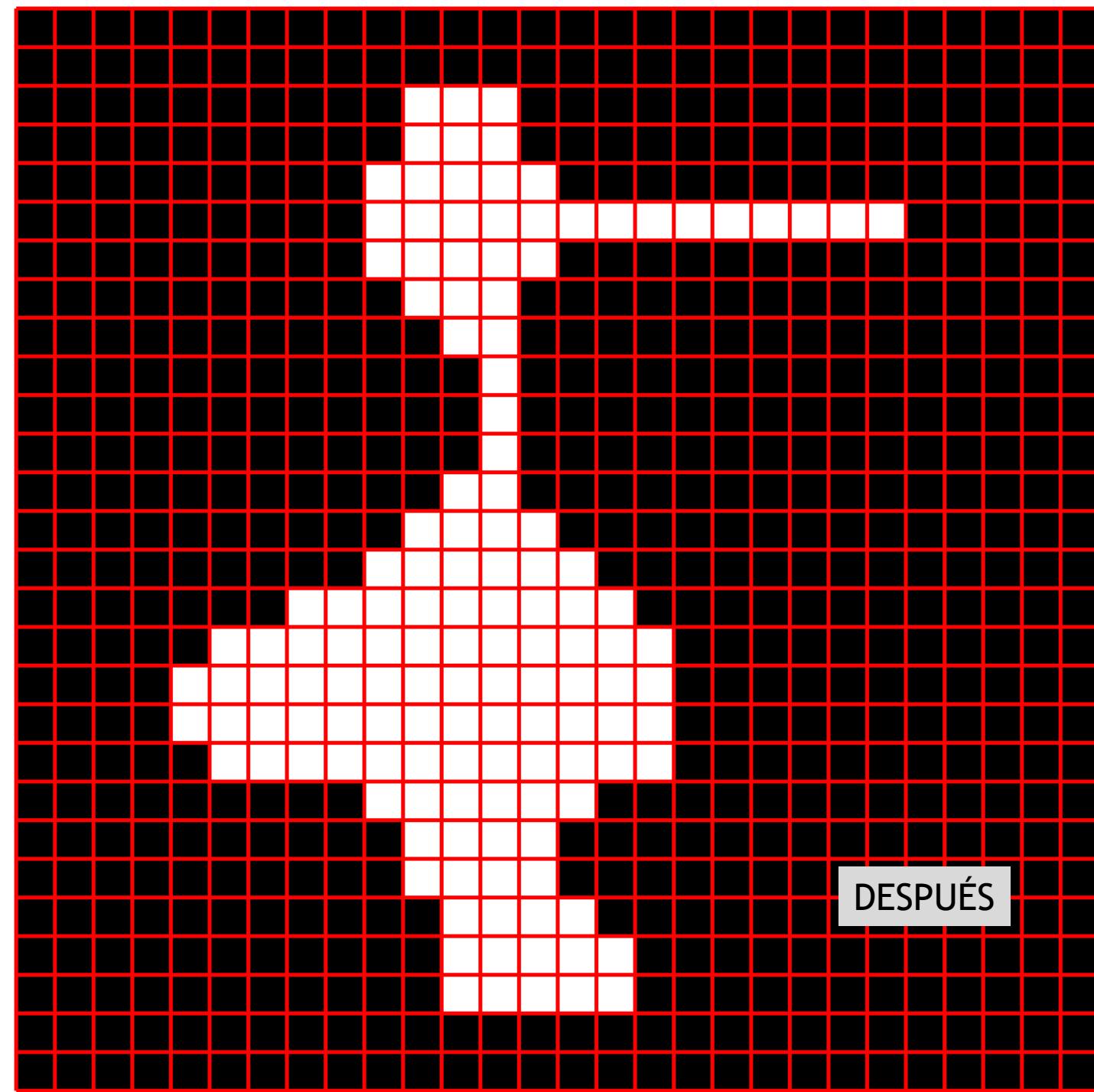


Ejemplo  
Imagen  
binaria  
 $28 \times 28$

Erosión con



DESPUÉS



# Dilatación y Erosión

$$Y = X \oplus S$$

- Dilatación: El output Y es ‘1’ si hay coincidencia en al menos un elemento de la estructura S con los elementos que cubre en X.
  - Esta operación equivale al máximo.

$$Y = X \ominus S$$

- Erosión: El output es ‘1’ sólo si hay coincidencia de todos los elementos de la estructura S con los elementos que cubre en X.
  - Esta operación equivale al mínimo.

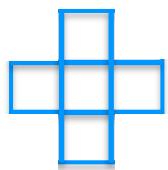
# Perímetro

Algoritmo simple

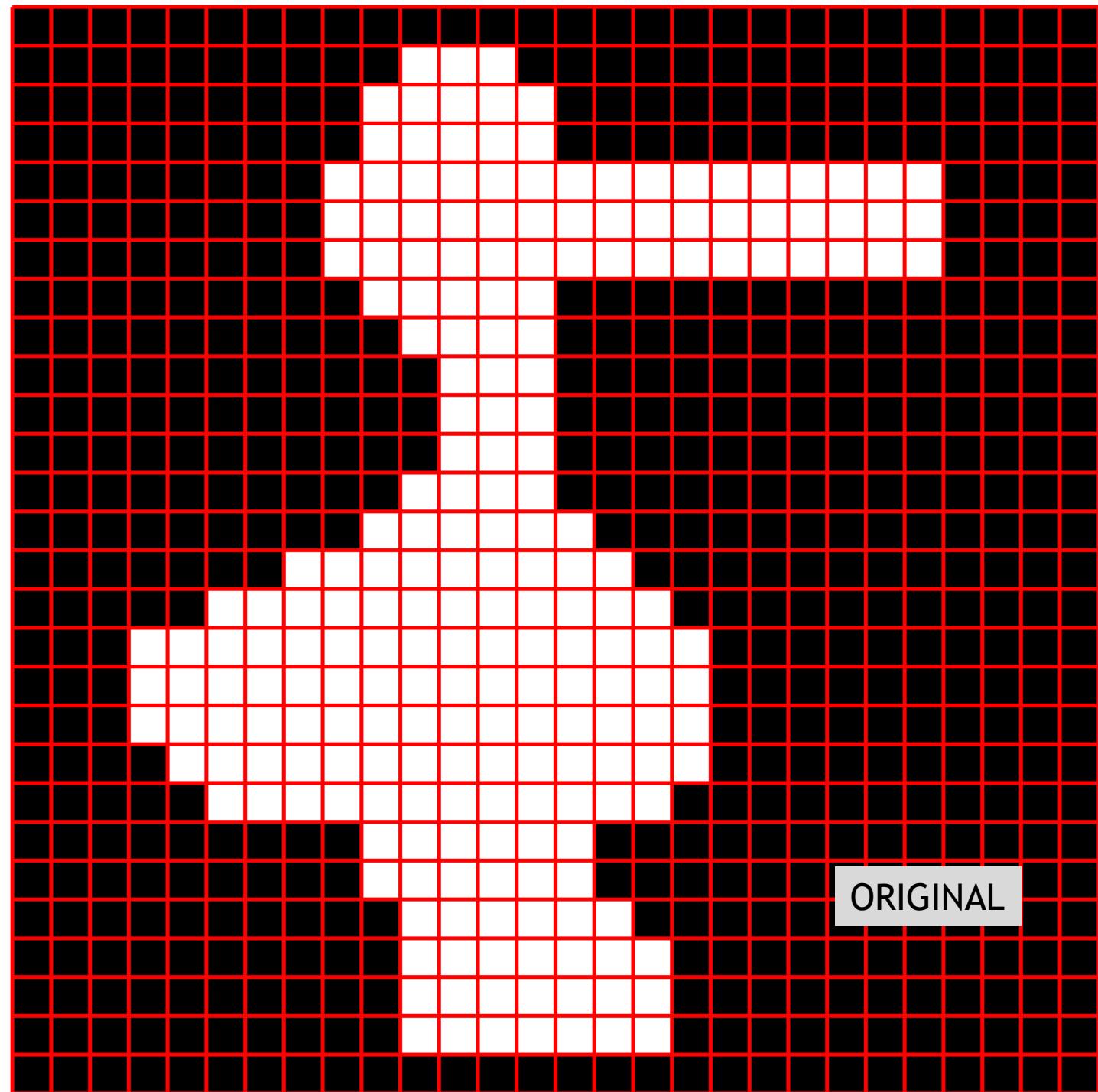
$$\mathbf{Y} = \mathbf{X} - (\mathbf{X} \odot \mathbf{S})$$

Ejemplo  
Imagen  
binaria  
 $28 \times 28$

Erosión con

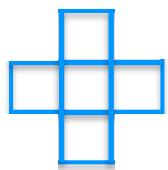


ORIGINAL

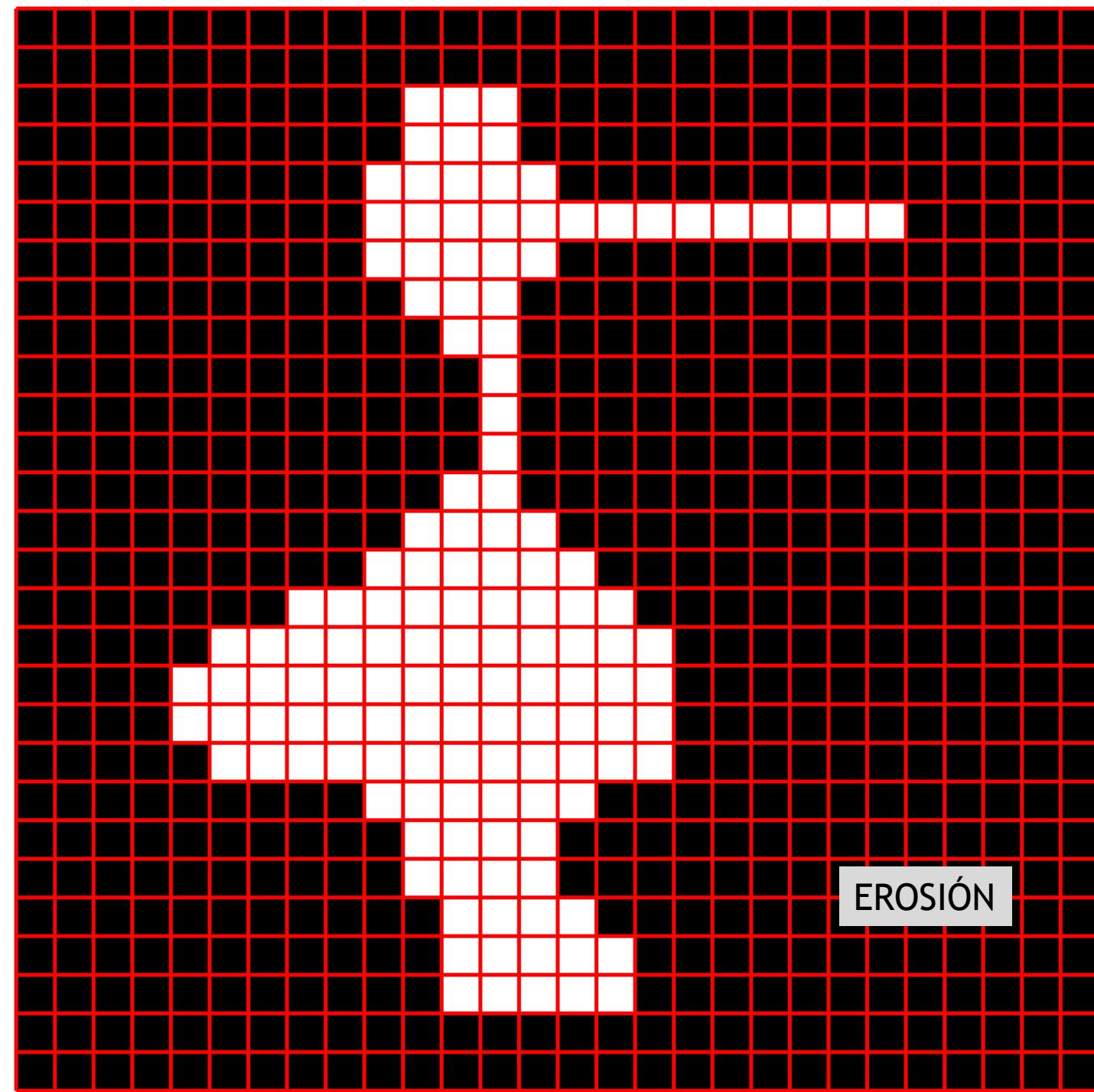


Ejemplo  
Imagen  
binaria  
 $28 \times 28$

Erosión con

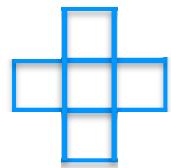


EROSIÓN



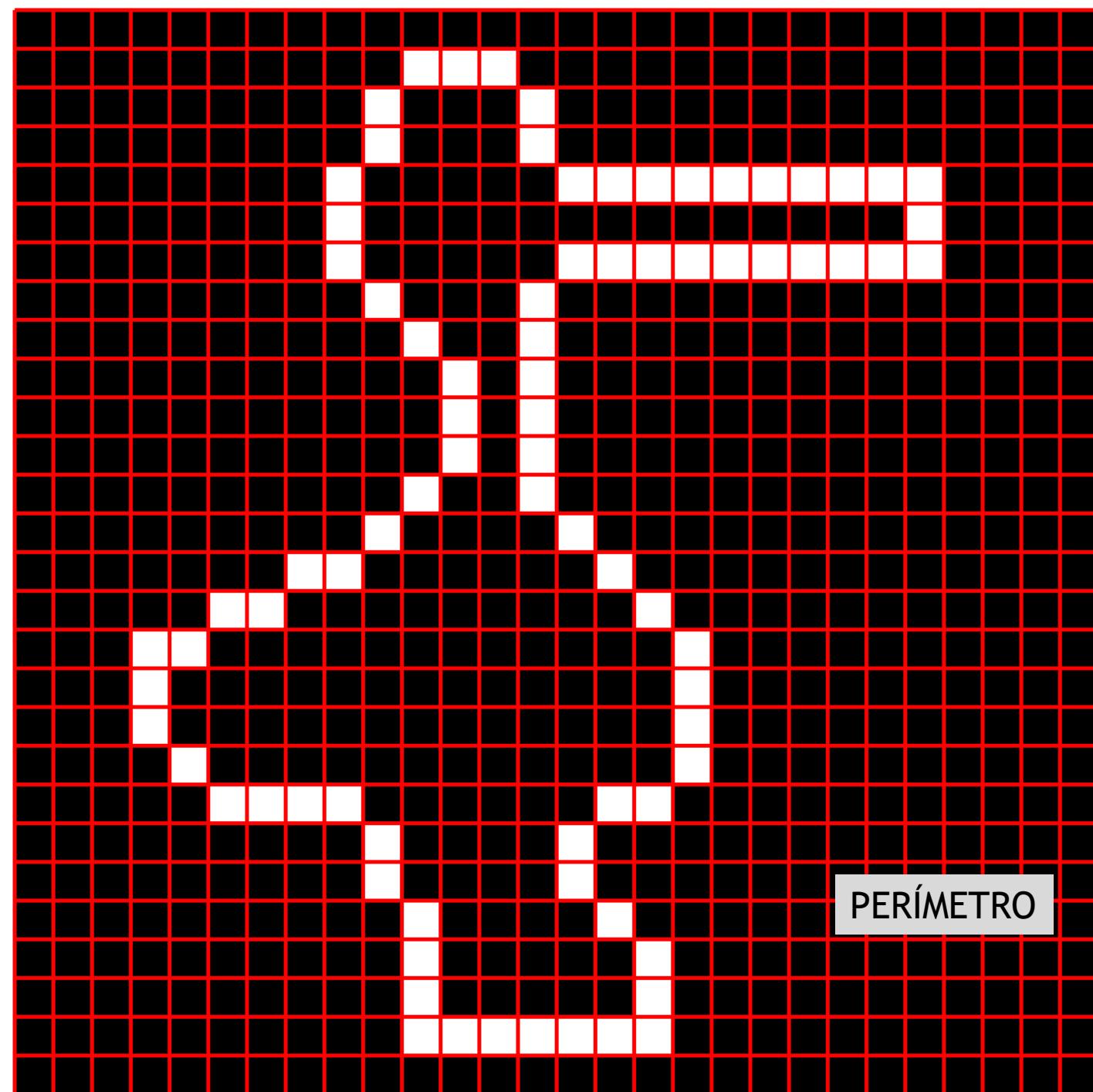
Ejemplo  
Imagen  
binaria  
 $28 \times 28$

Erosión con



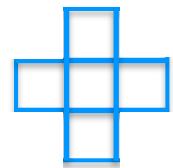
$$Y = X - (X \odot S)$$

PERÍMETRO



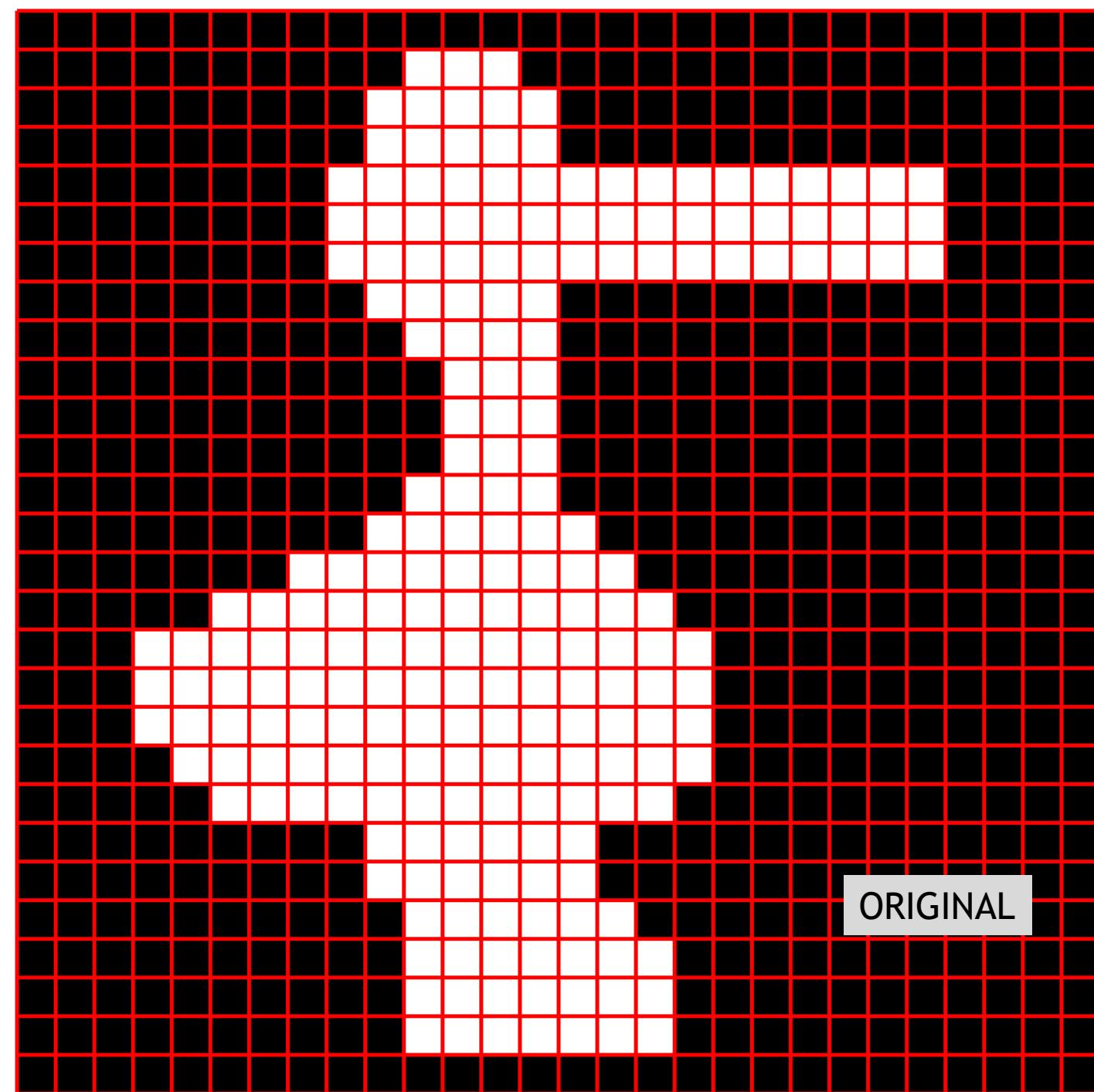
Ejemplo  
Imagen  
binaria  
 $28 \times 28$

Erosión con

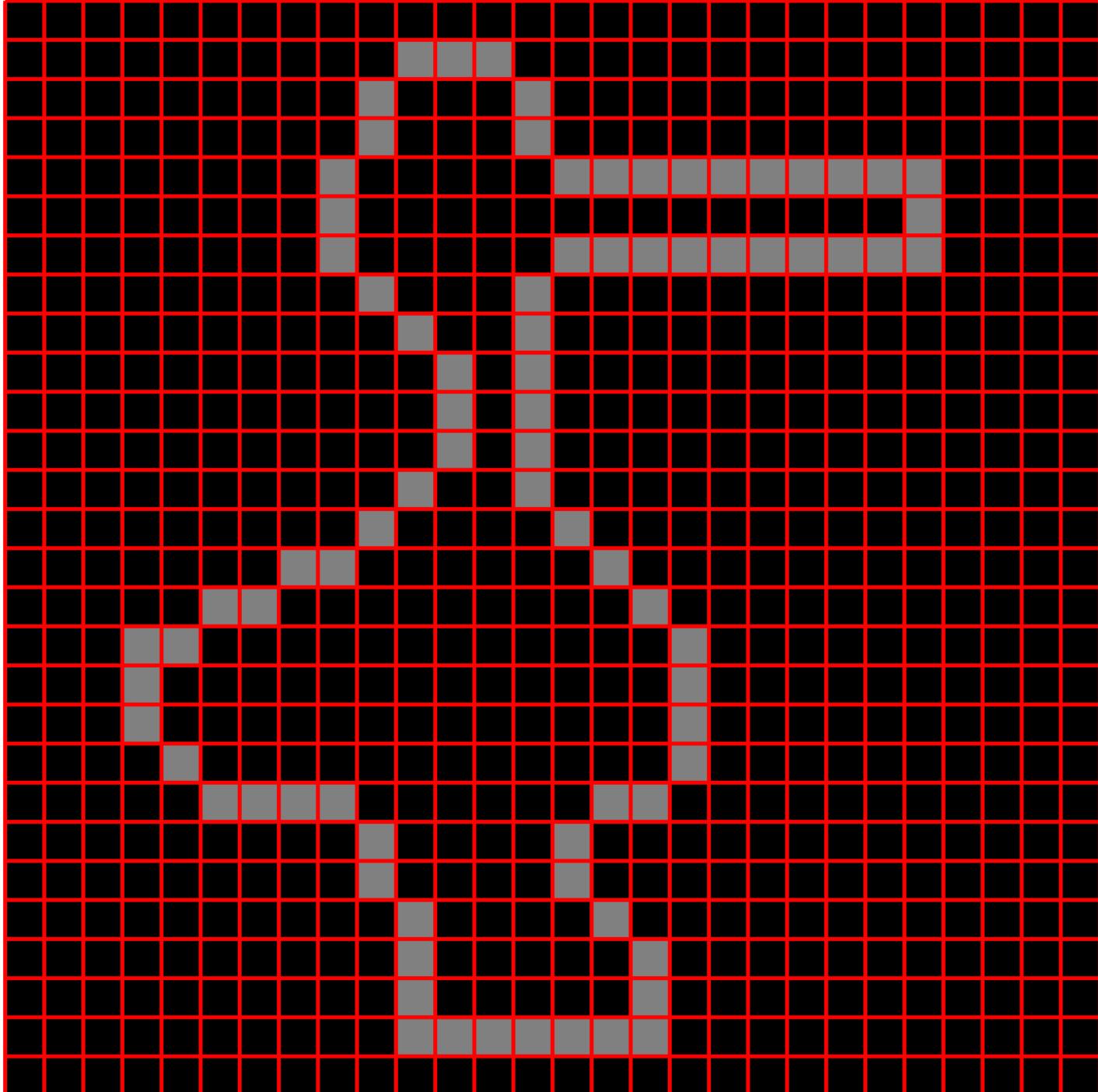


$$Y = X - (X \odot S)$$

ORIGINAL



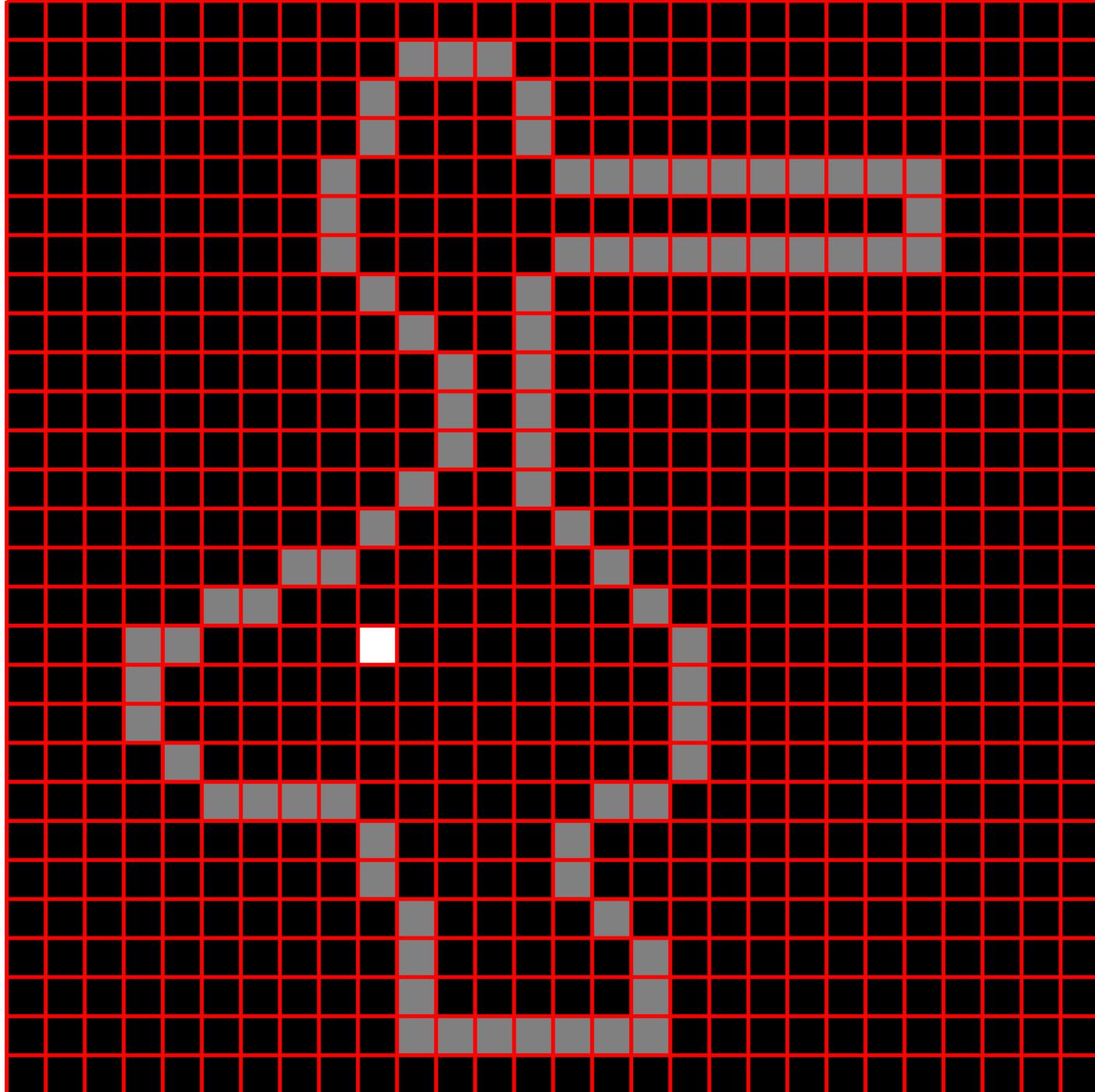
# Relleno de Región



Se desea  
rellenar el  
interior

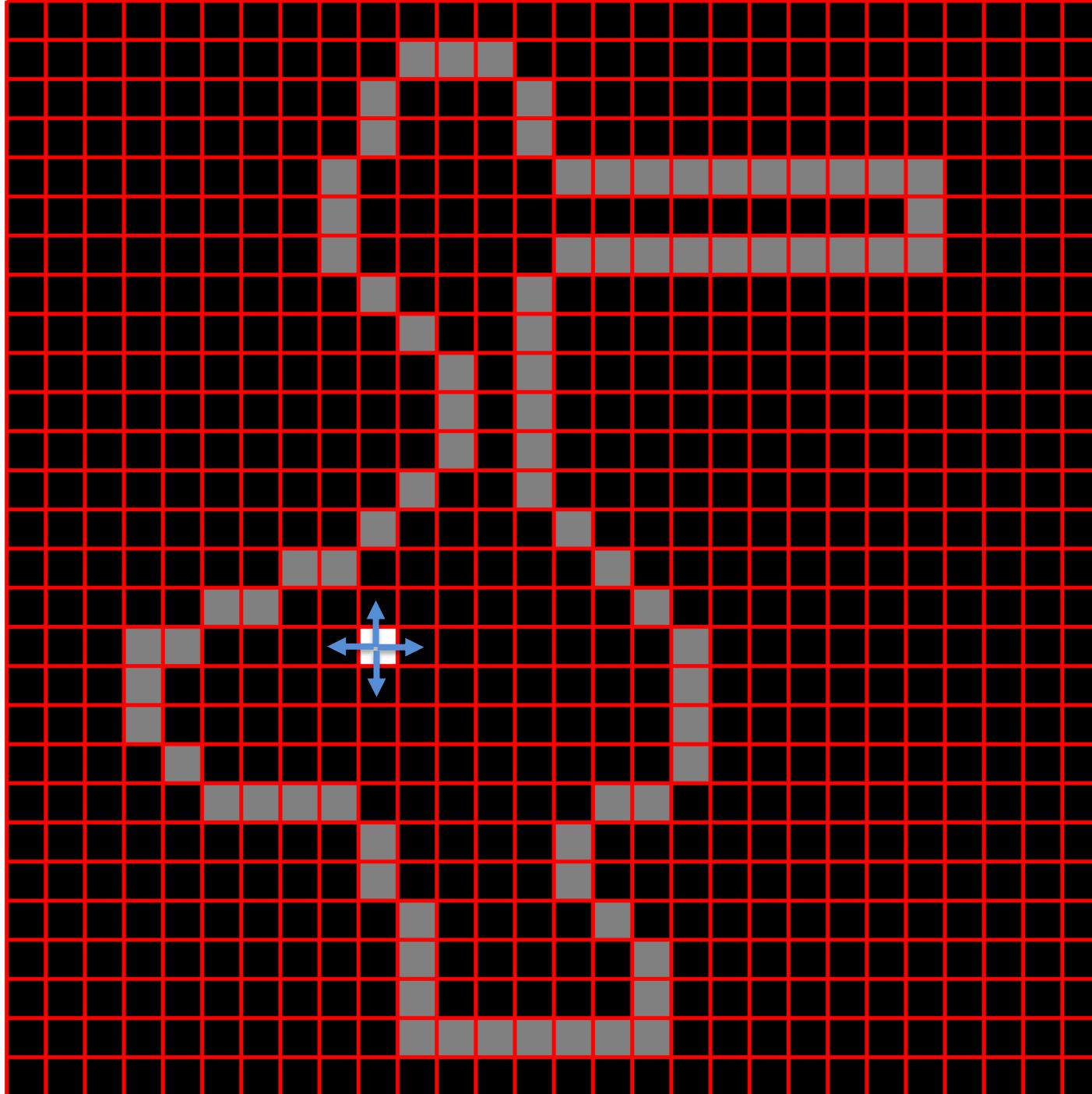
Se desea  
rellenar el  
interior

1. Escogemos un pixel semilla



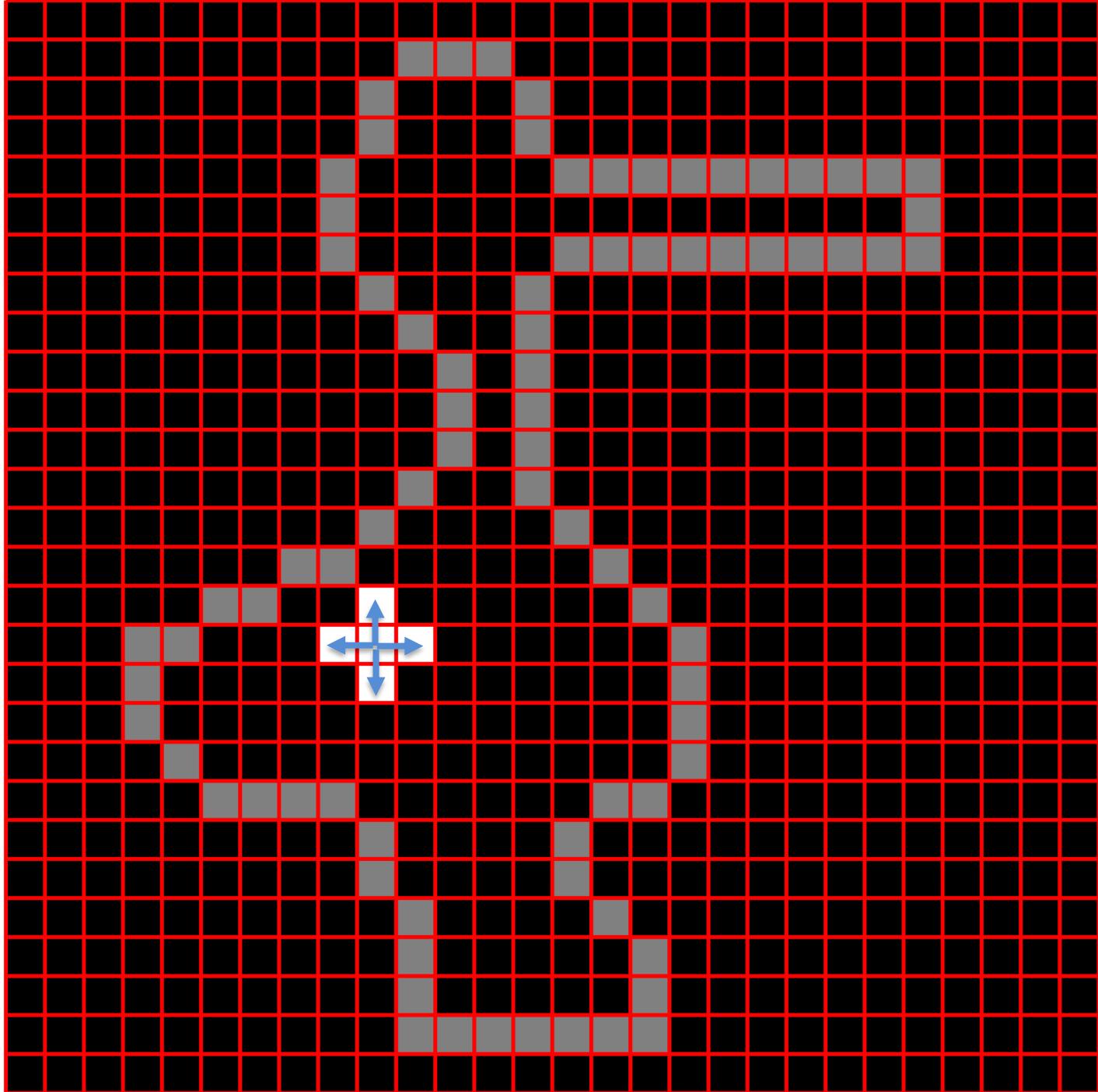
Se desea  
rellenar el  
interior

1. Escogemos un pixel semilla
2. Hacemos crecer en 4 direcciones



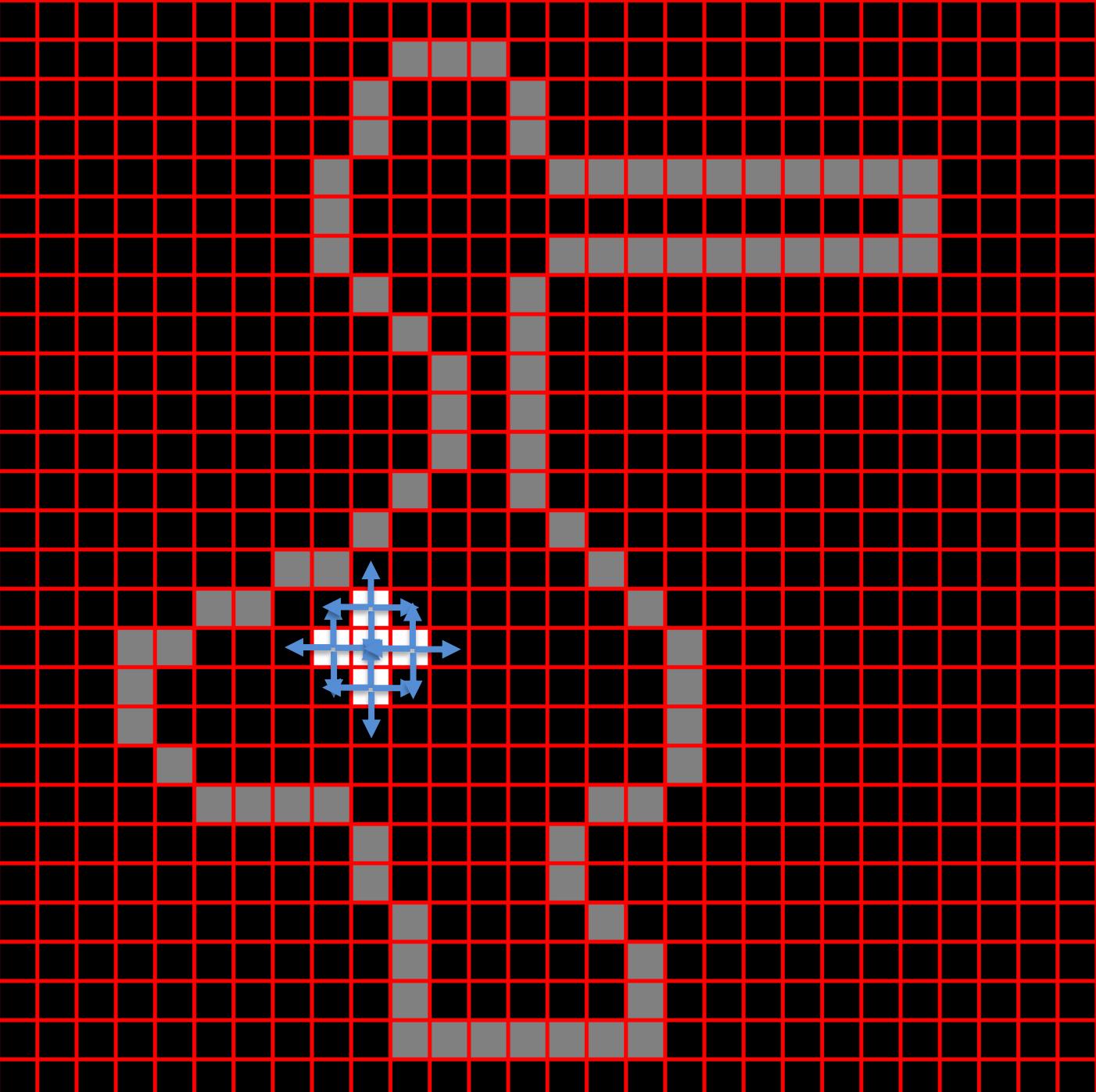
Se desea  
rellenar el  
interior

1. Escogemos un pixel semilla
2. Hacemos crecer en 4 direcciones
3. Repetimos en forma iterativa respetando borde (gris)



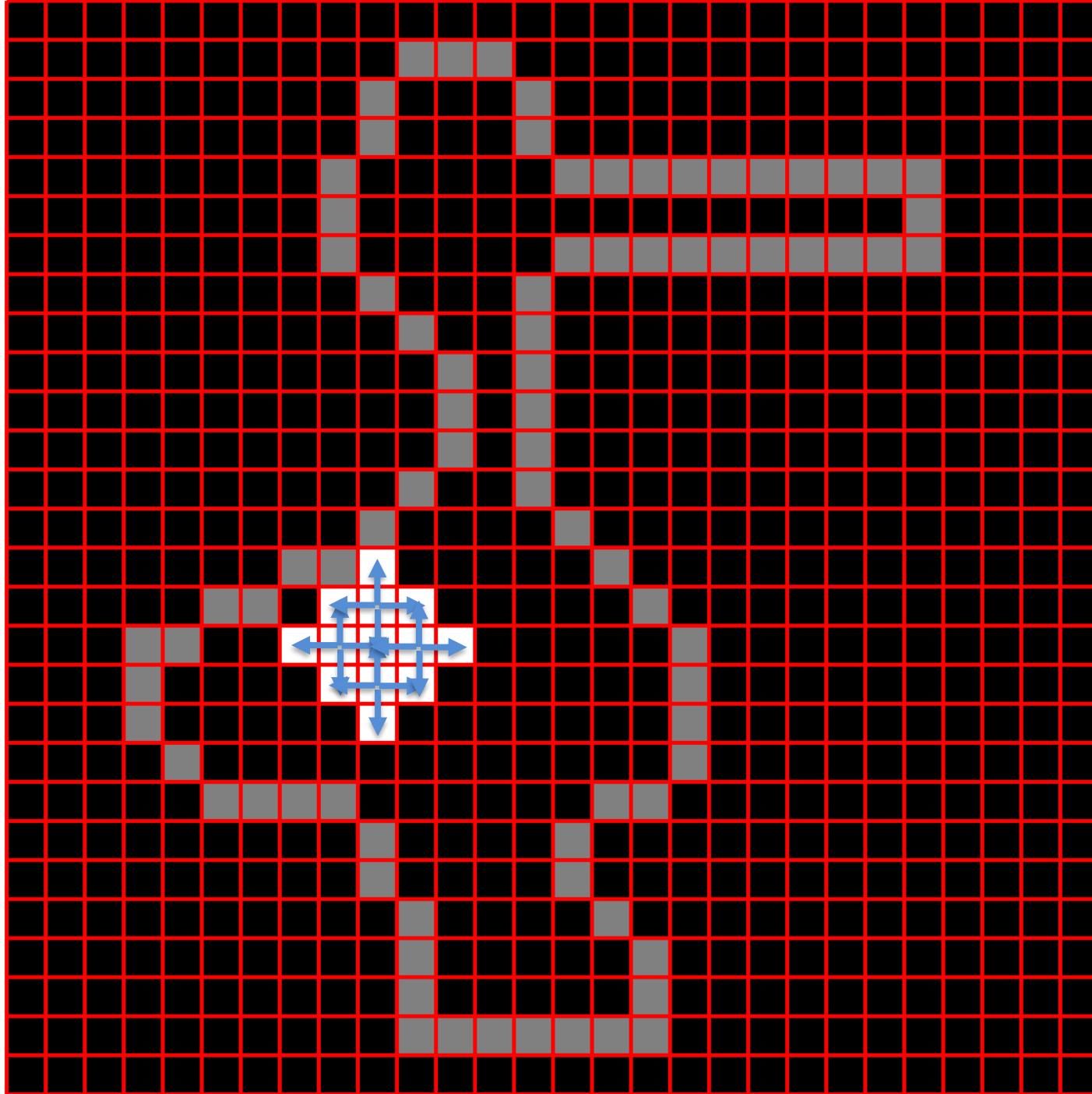
Se desea  
rellenar el  
interior

1. Escogemos un pixel semilla
2. Hacemos crecer en 4 direcciones
3. Repetimos en forma iterativa respetando borde (gris)



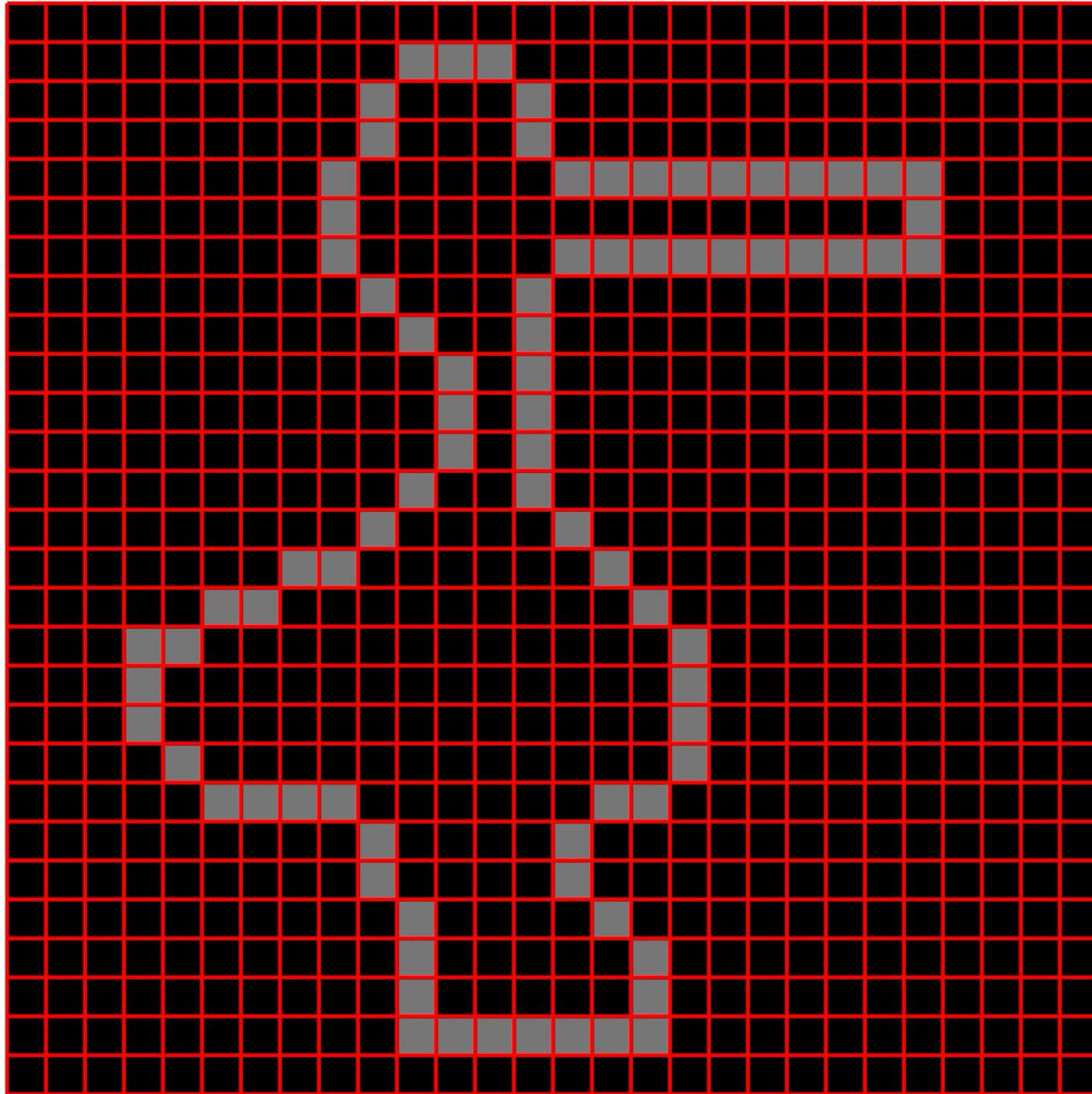
Se desea  
rellenar el  
interior

1. Escogemos un pixel semilla
2. Hacemos crecer en 4 direcciones
3. Repetimos en forma iterativa respetando borde (gris)



Se desea  
rellenar el  
interior

1. Escogemos un pixel semilla
2. Hacemos crecer en 4 direcciones
3. Repetimos en forma iterativa respetando borde (gris)



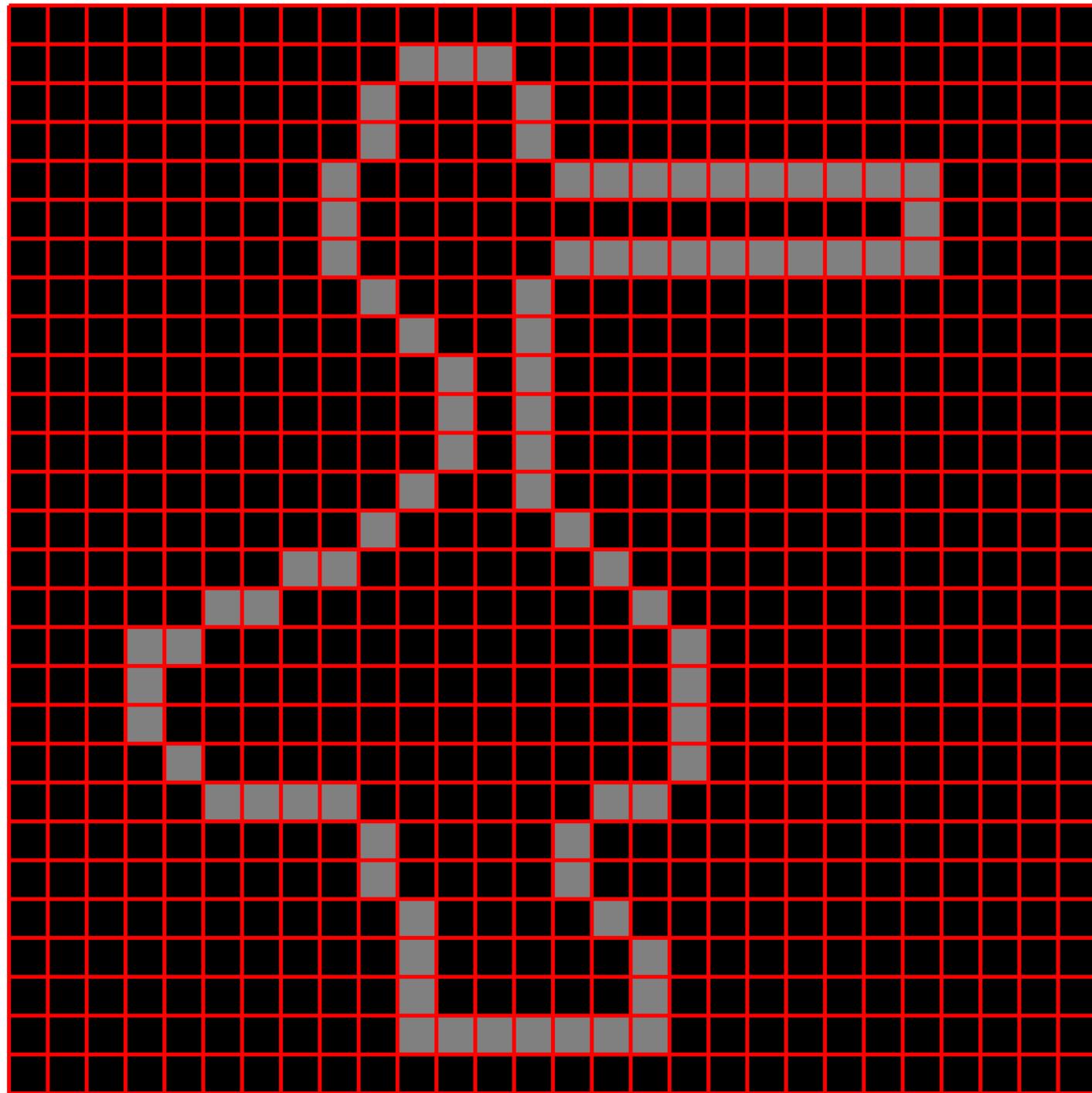
# Algoritmo

E = bordes

X = zeros(28,28)

(i,j) = semilla

X(i,j) = 1



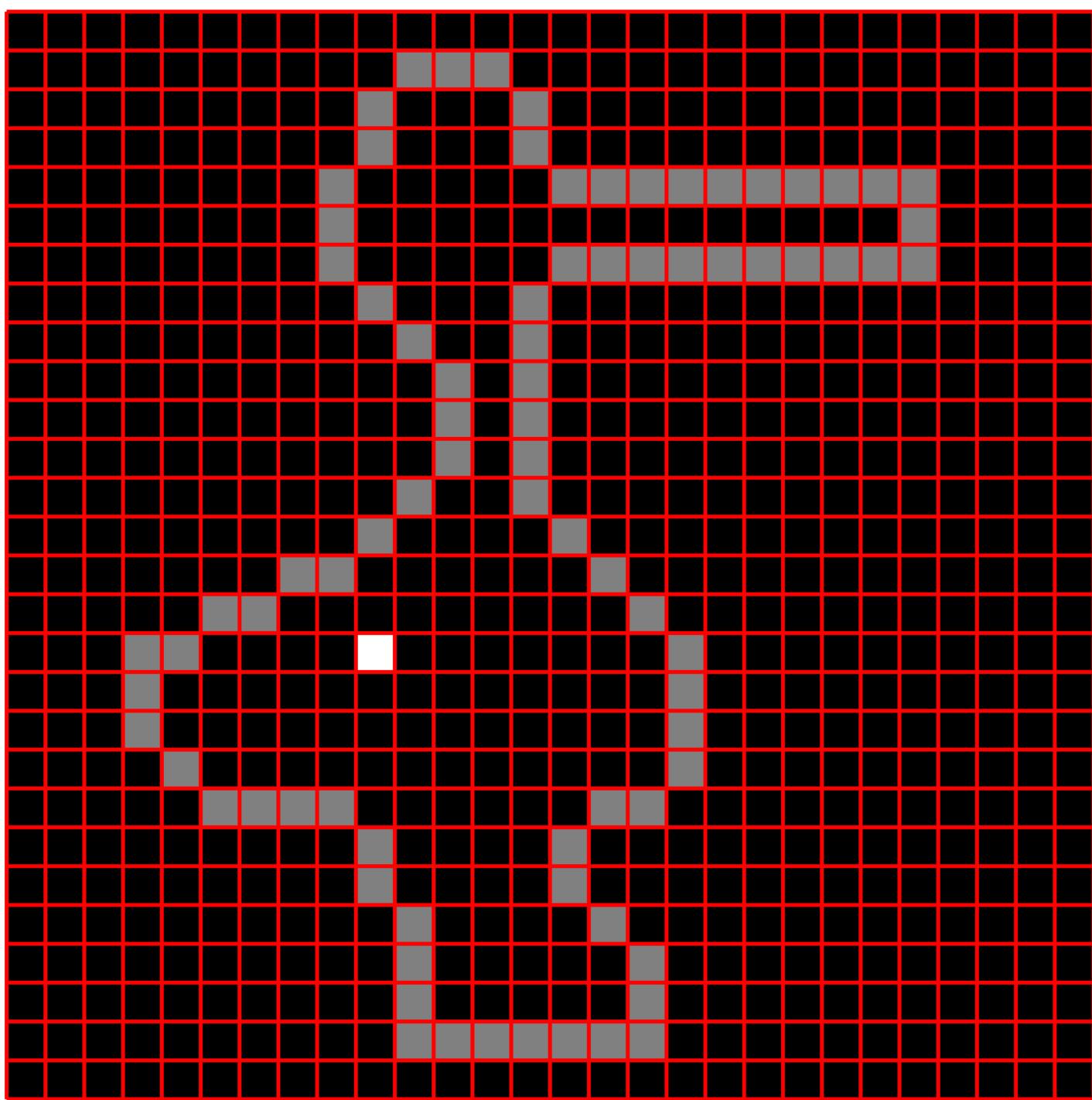
## Algoritmo

E = bordes

X = zeros(28,28)

(i,j) = semilla

X(i,j) = 1



## Algoritmo

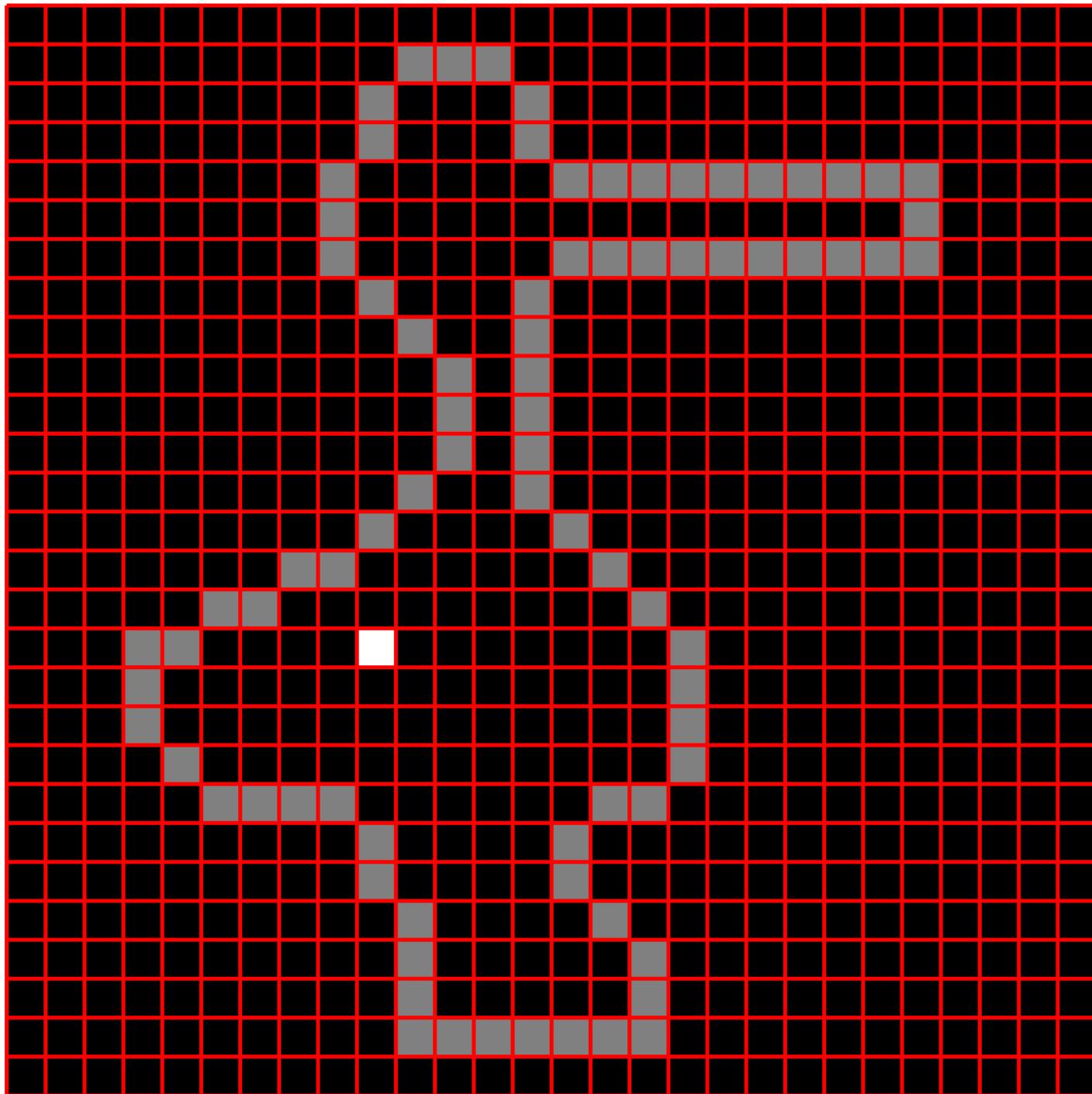
E = bordes

X = zeros(28,28)

(i,j) = semilla

X(i,j) = 1

$$S = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$



## Algoritmo

$E = \text{bordes}$

$X = \text{zeros}(28,28)$

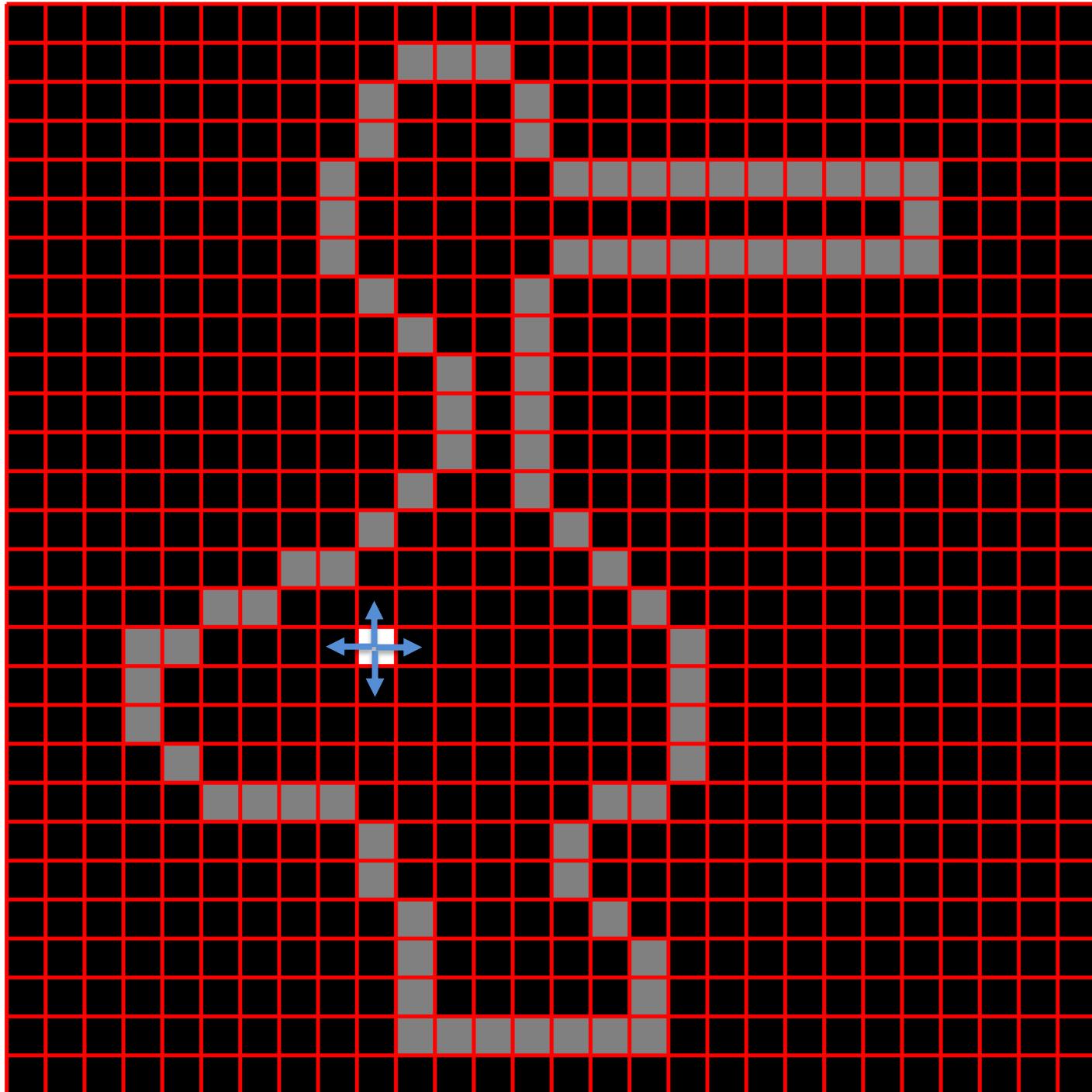
$(i,j) = \text{semilla}$

$X(i,j) = 1$

$$S = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$X = X \odot S$

↑  
Dilatación



## Algoritmo

$E = \text{bordes}$

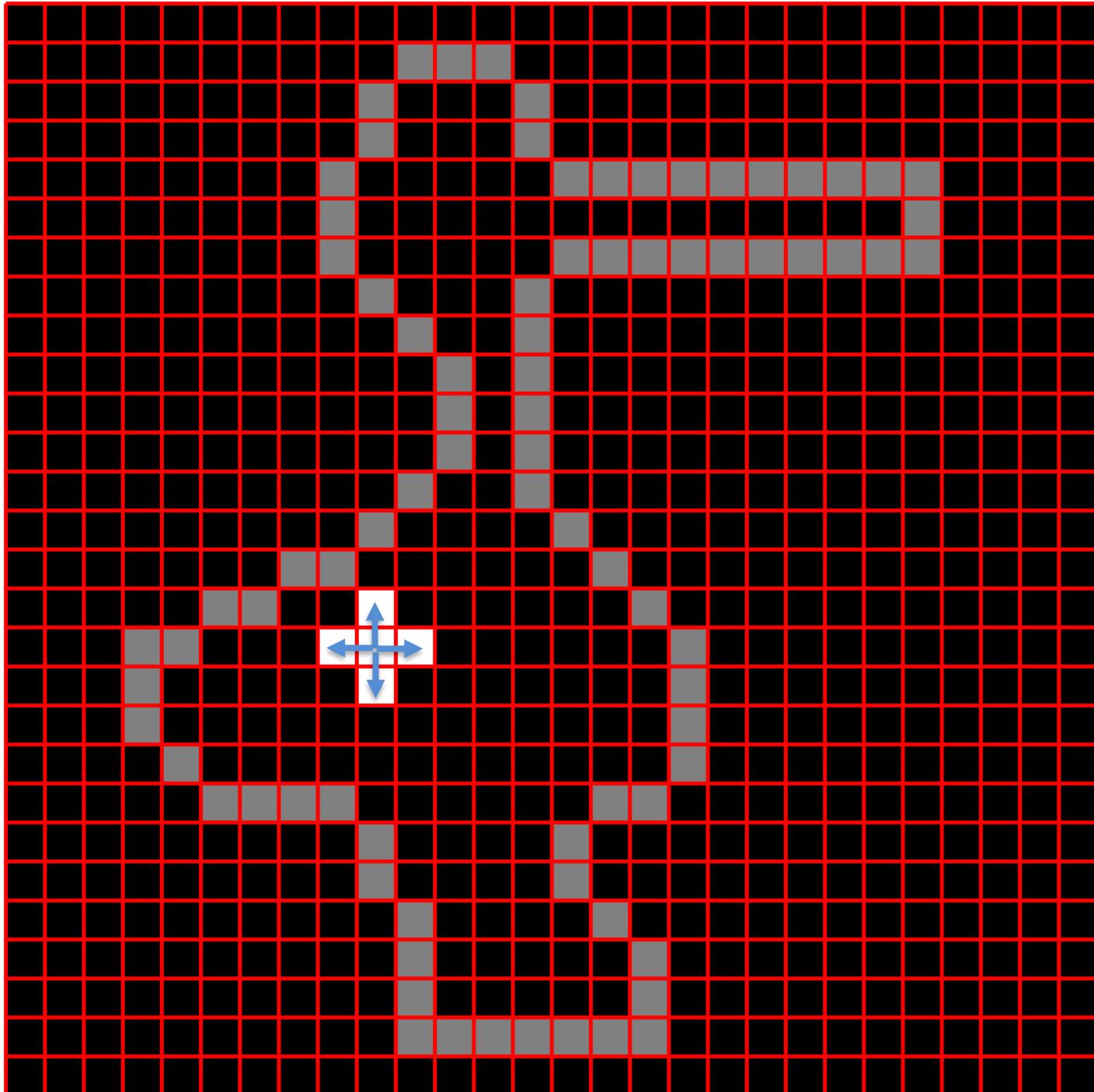
$X = \text{zeros}(28,28)$

$(i,j) = \text{semilla}$

$X(i,j) = 1$

$$S = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$X = X \odot S$$



# Algoritmo

$E$  = bordes

$X = \text{zeros}(28,28)$

$(i,j) = \text{semilla}$

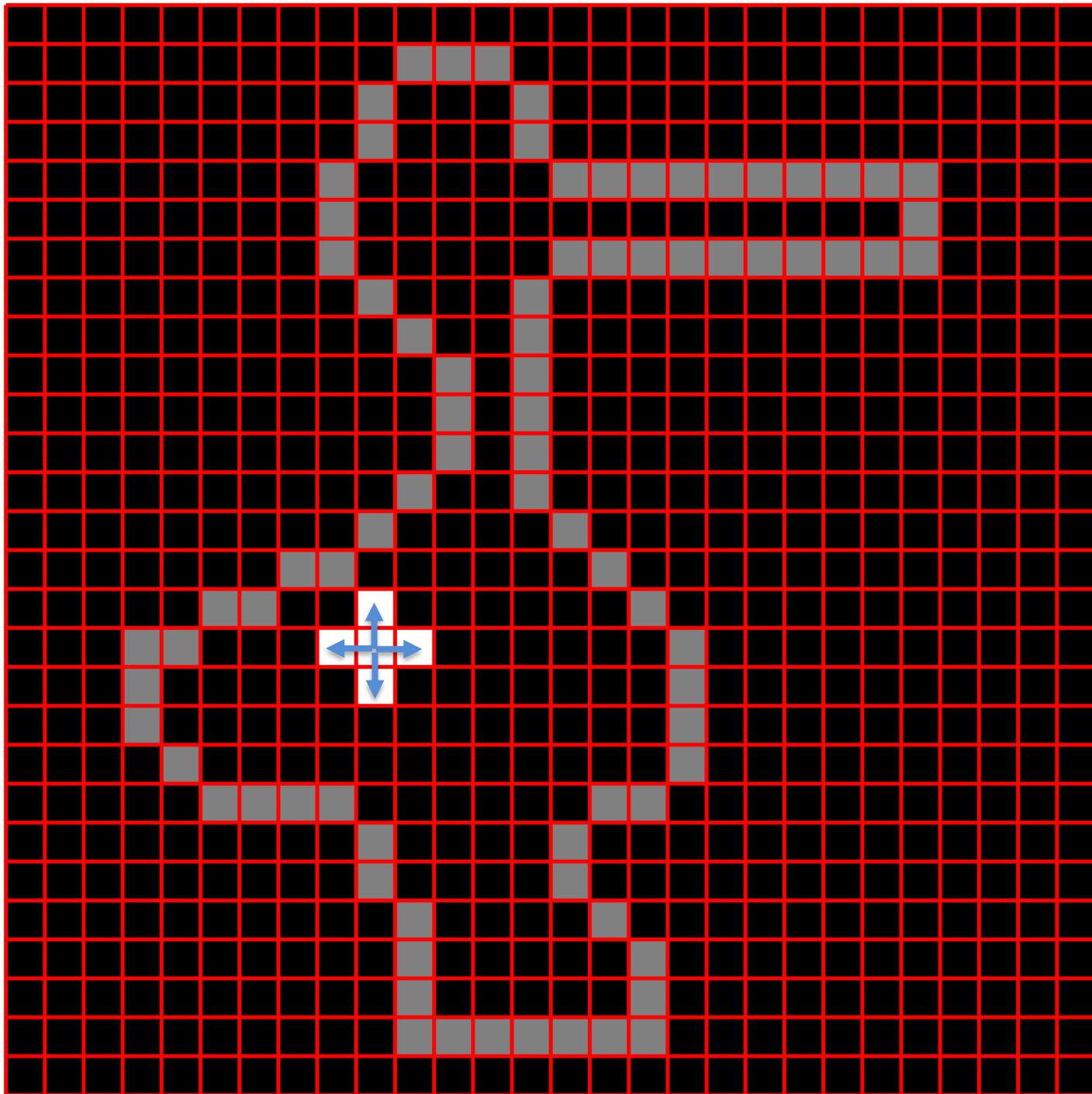
$X(i,j) = 1$

$$S = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

NOT

$$X = (X \oplus S) \cdot \bar{E}$$

AND



## Algoritmo

$E = \text{bordes}$

$X = \text{zeros}(28,28)$

$(i,j) = \text{semilla}$

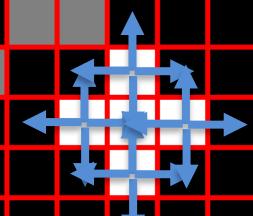
$X(i,j) = 1$

$$S = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$X_{k+1} = (X_k \oplus S) \bullet \bar{E}$$

parar cuando:

$$X_{k+1} = X_k$$



## Algoritmo

$E = \text{bordes}$

$X = \text{zeros}(28,28)$

$(i,j) = \text{semilla}$

$X(i,j) = 1$

$$S = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$X_{k+1} = (X_k \oplus S) \bullet \bar{E}$$

parar cuando:

$$X_{k+1} = X_k$$

# Cierre

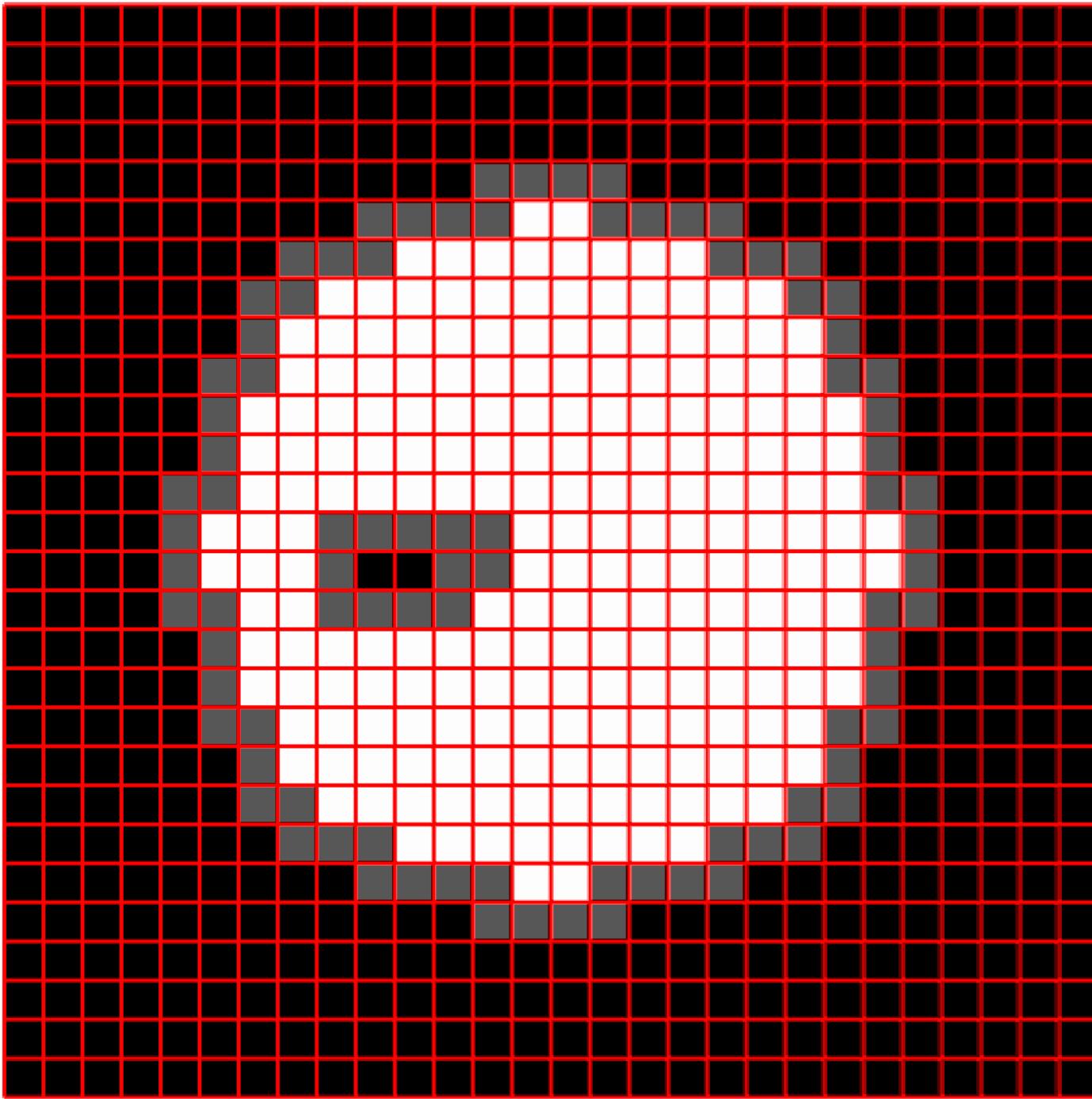
$$Y = X \bullet S$$

|            |            |  
Output      Input      Estructura

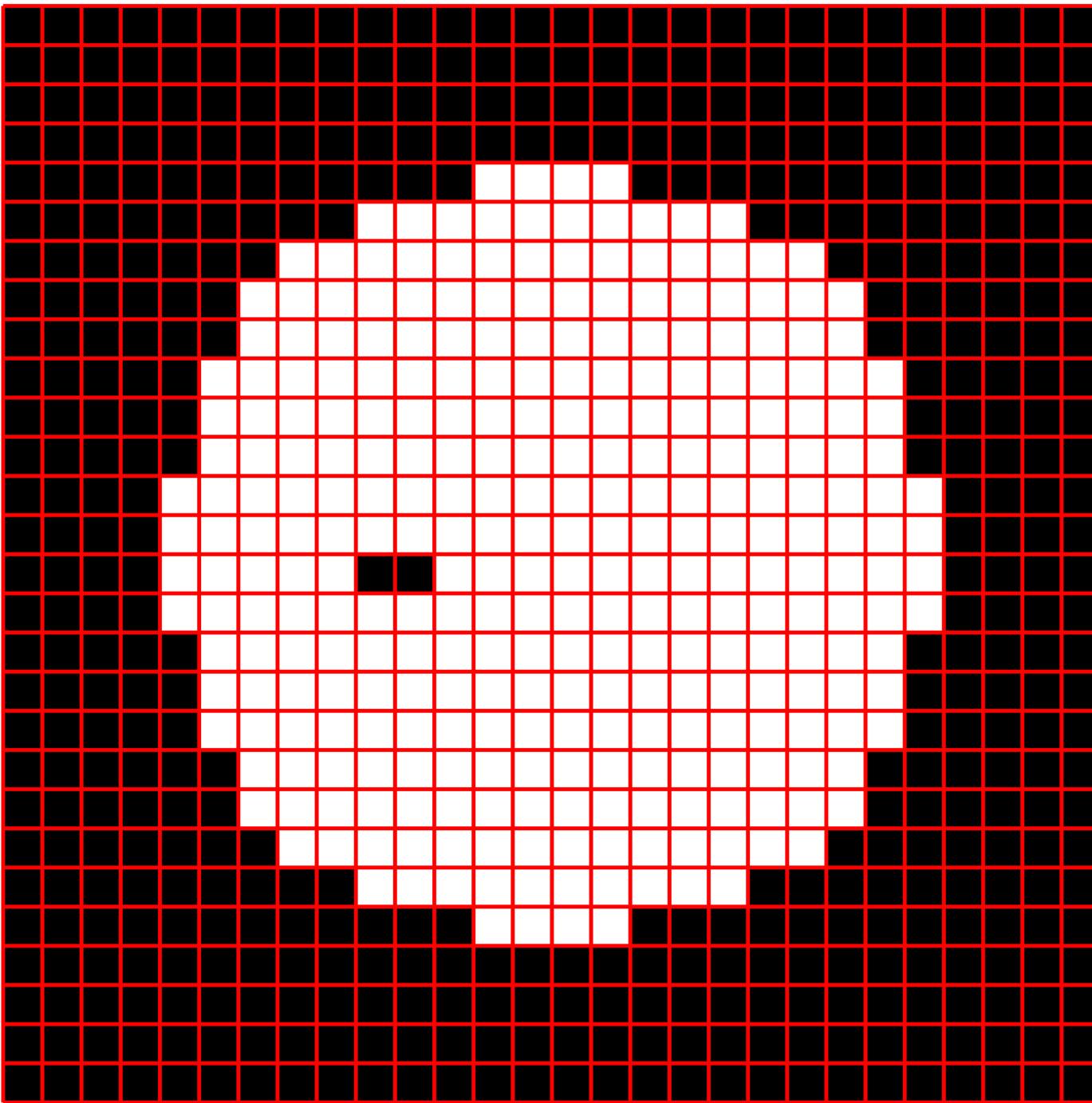
# Ejemplo Imagen binaria $28 \times 28$

Se desea  
rellenar el  
agujero.

¿Qué pasaría si  
hacemos una  
**dilatación** con  
una estructura  
 $3 \times 3$ ?



# Ejemplo Imagen binaria $28 \times 28$



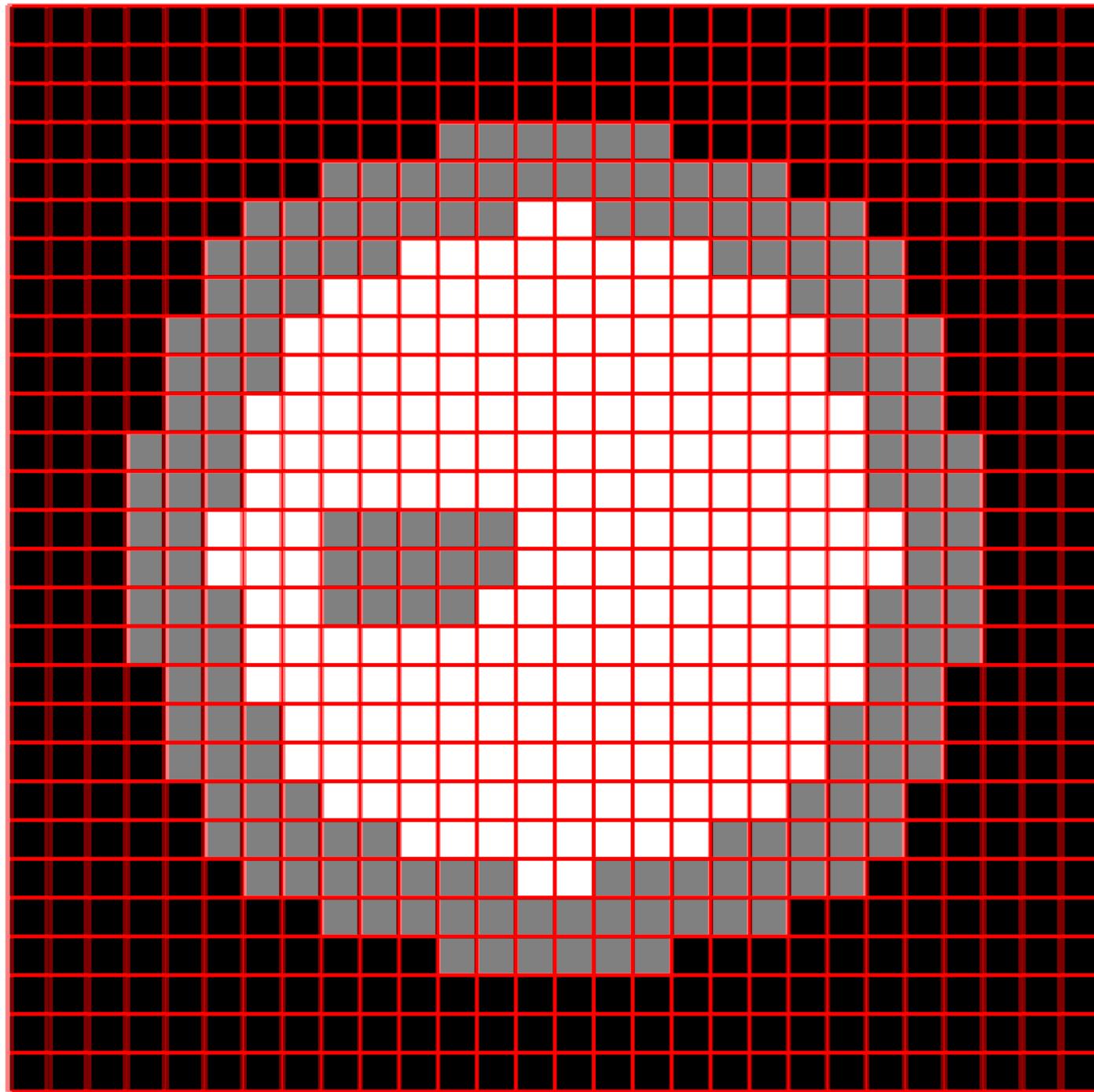
Se desea  
rellenar el  
agujero.

La estructura  
 $3 \times 3$  no es  
suficiente.

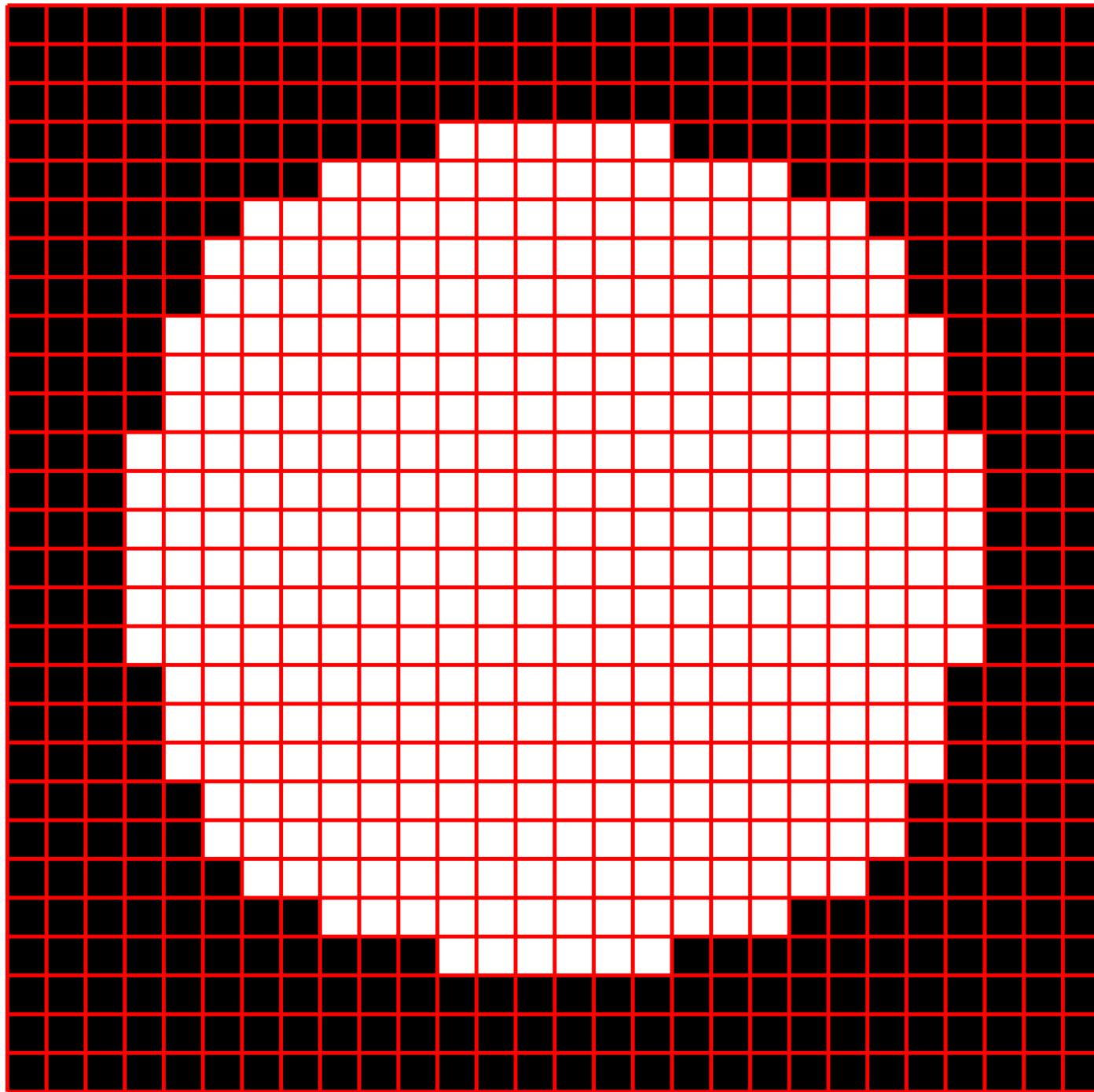
# Ejemplo Imagen binaria $28 \times 28$

Se desea  
rellenar el  
agujero.

¿Qué pasaría si  
hacemos una  
**dilatación** con  
una estructura  
 $5 \times 5$ ?



# Ejemplo Imagen binaria $28 \times 28$



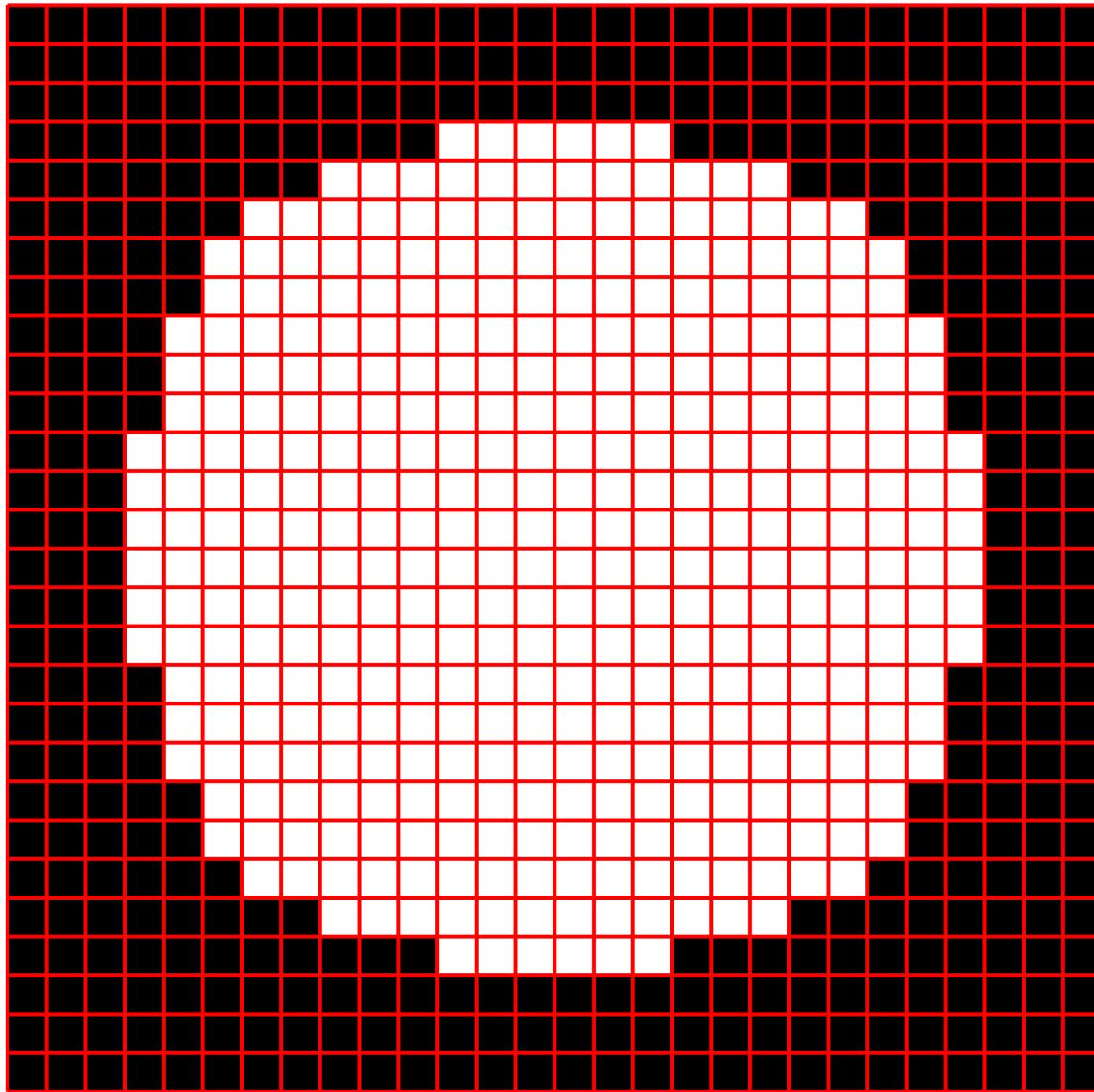
Se desea  
rellenar el  
agujero.

Se rellena el  
agujero pero se  
dilata demasiado.

# Ejemplo Imagen binaria $28 \times 28$

Se desea  
rellenar el  
agujero.

¿Qué pasaría si  
después de la  
**dilatación**  
hacemos una  
**erosión** con una  
estructura  $5 \times 5$ ?

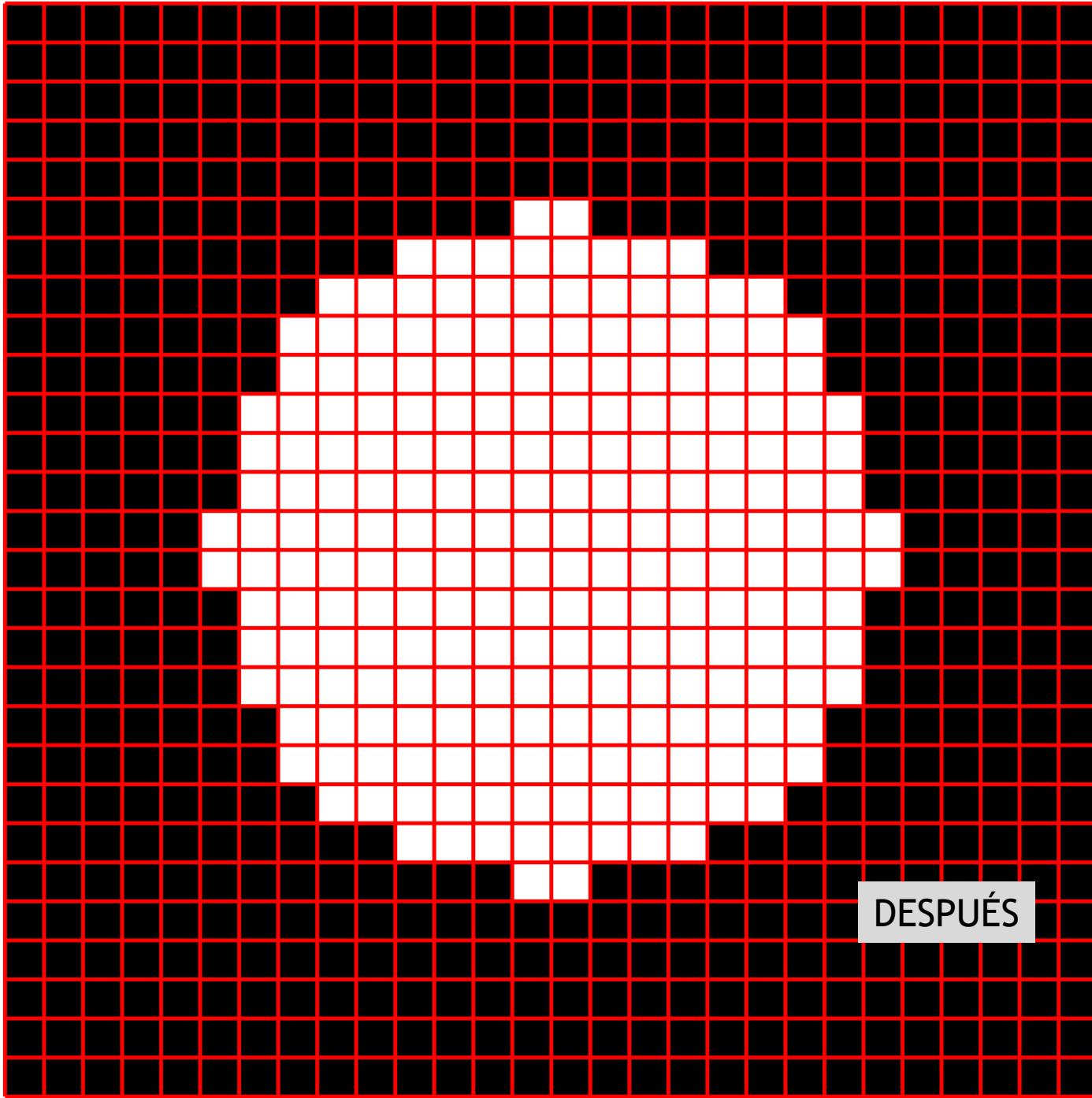


# Ejemplo Imagen binaria $28 \times 28$

Se desea  
rellenar el  
agujero.

Bien! Se rellena  
y se mantiene el  
tamaño.

DESPUÉS

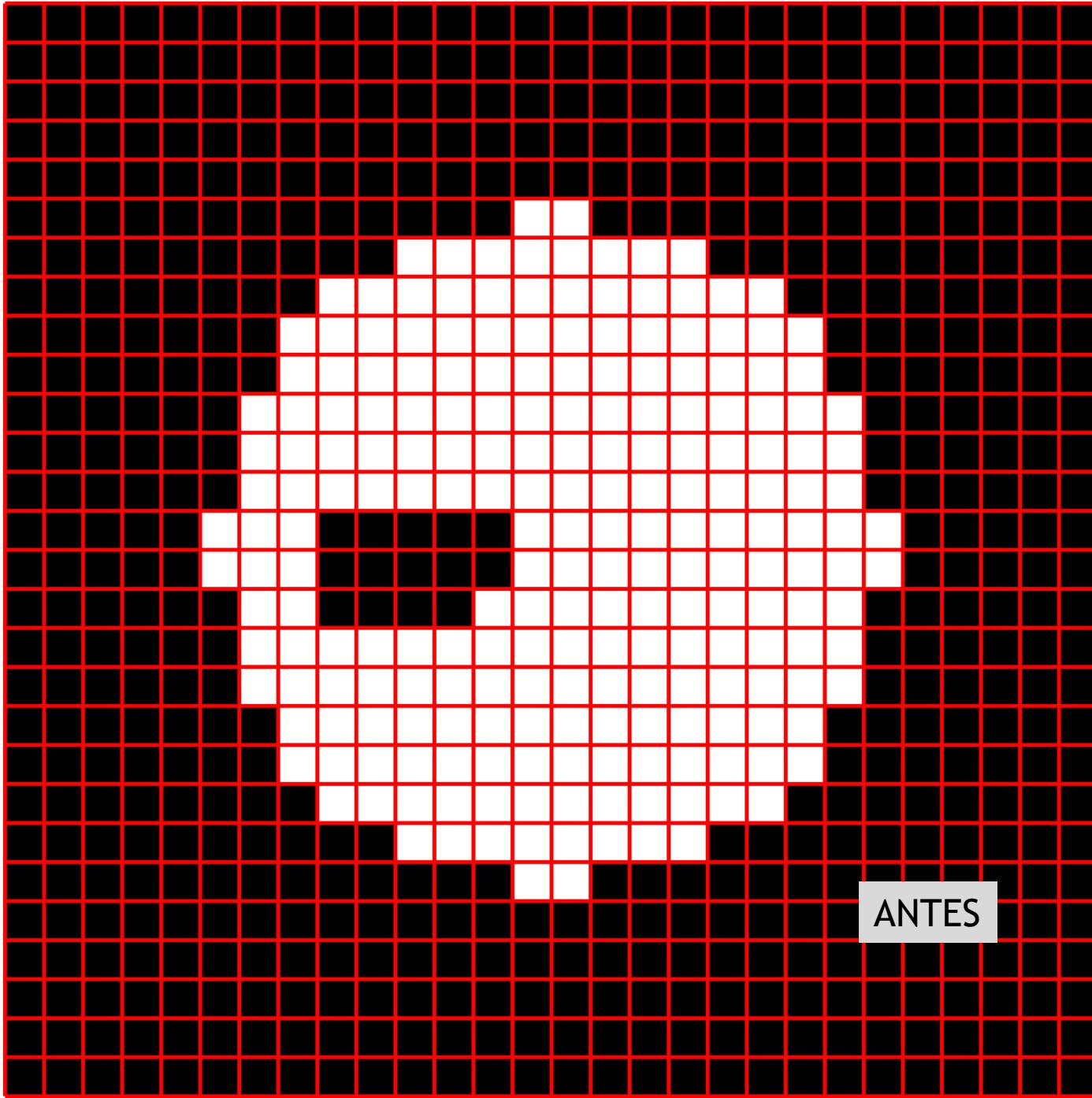


# Ejemplo Imagen binaria $28 \times 28$

Se desea  
rellenar el  
agujero.

Bien! Se rellena  
y se mantiene el  
tamaño.

ANTES



# Cierre

$$\mathbf{Y} = \mathbf{X} \bullet \mathbf{S} = (\mathbf{X} \oplus \mathbf{S}) \ominus \mathbf{S}$$

|      |      |  
Output   Input   Estructura

1. Dilatación
2. Erosión

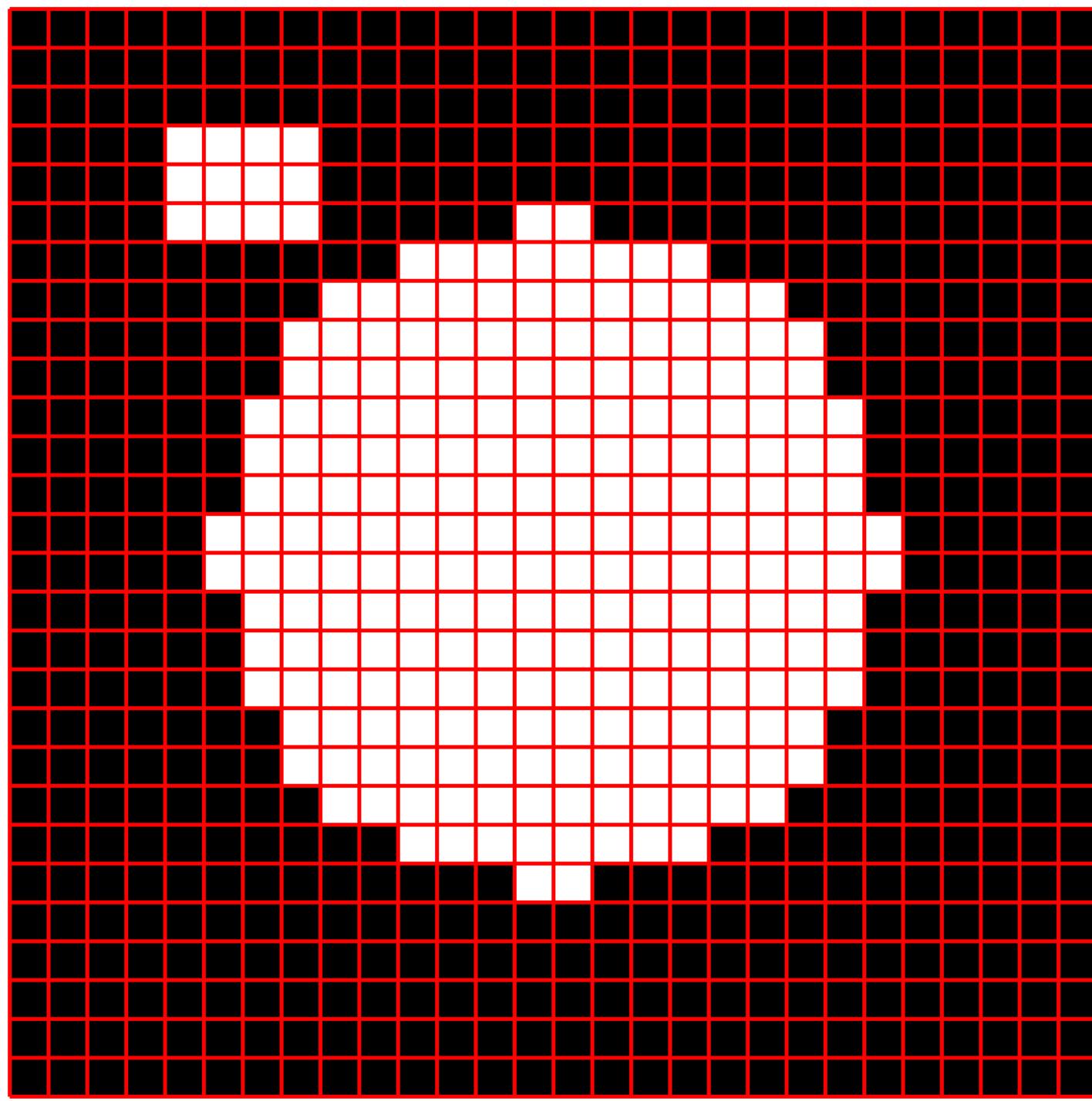
# Apertura

$$\mathbf{Y} = \mathbf{X} \circ \mathbf{S}$$

|           |           |  
Output   Input   Estructura

# Ejemplo Imagen binaria $28 \times 28$

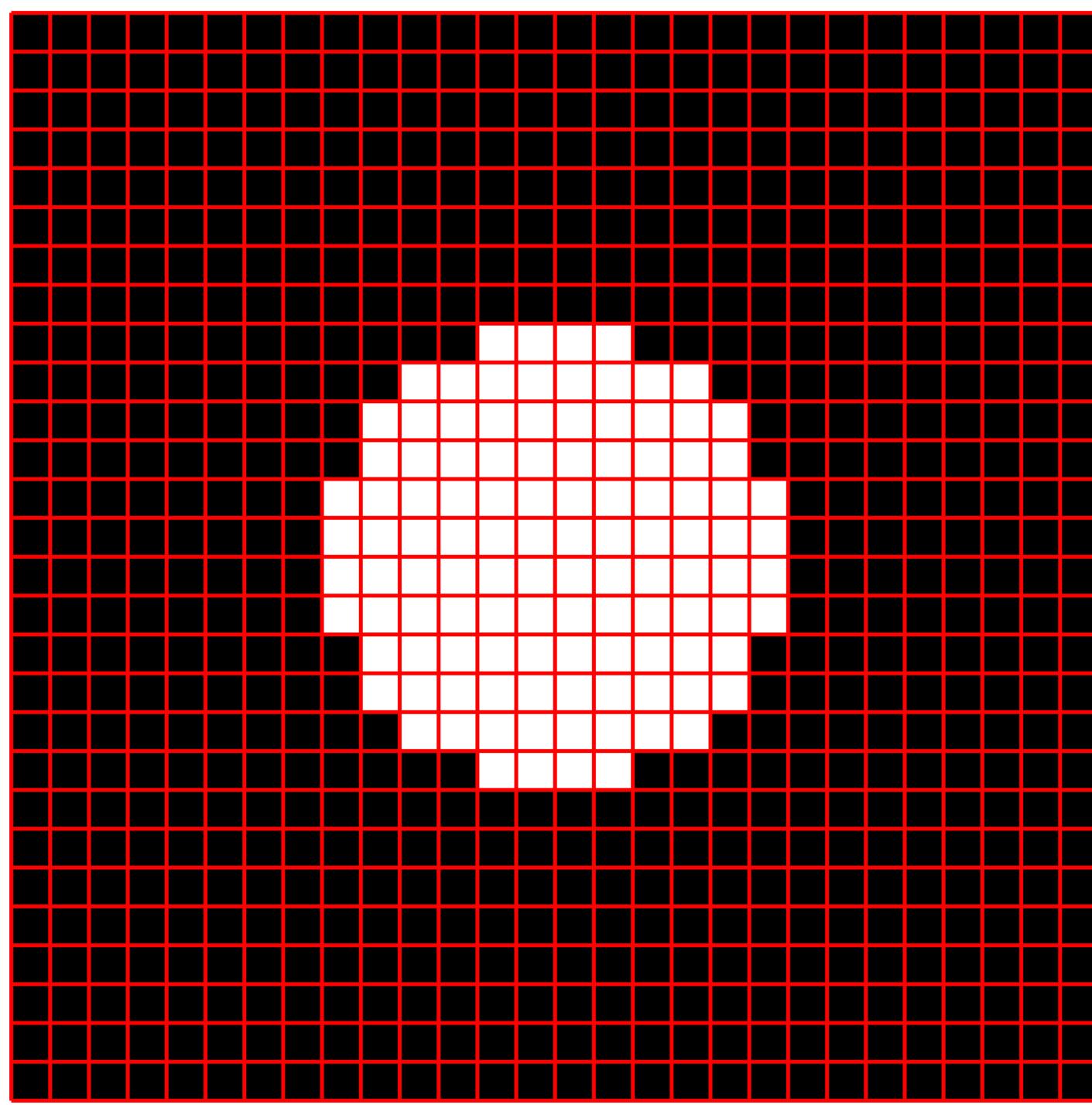
Se desea  
eliminar  
rectángulo.



# Ejemplo Imagen binaria $28 \times 28$

Se desea  
eliminar  
rectángulo.

Erosión 5x5.

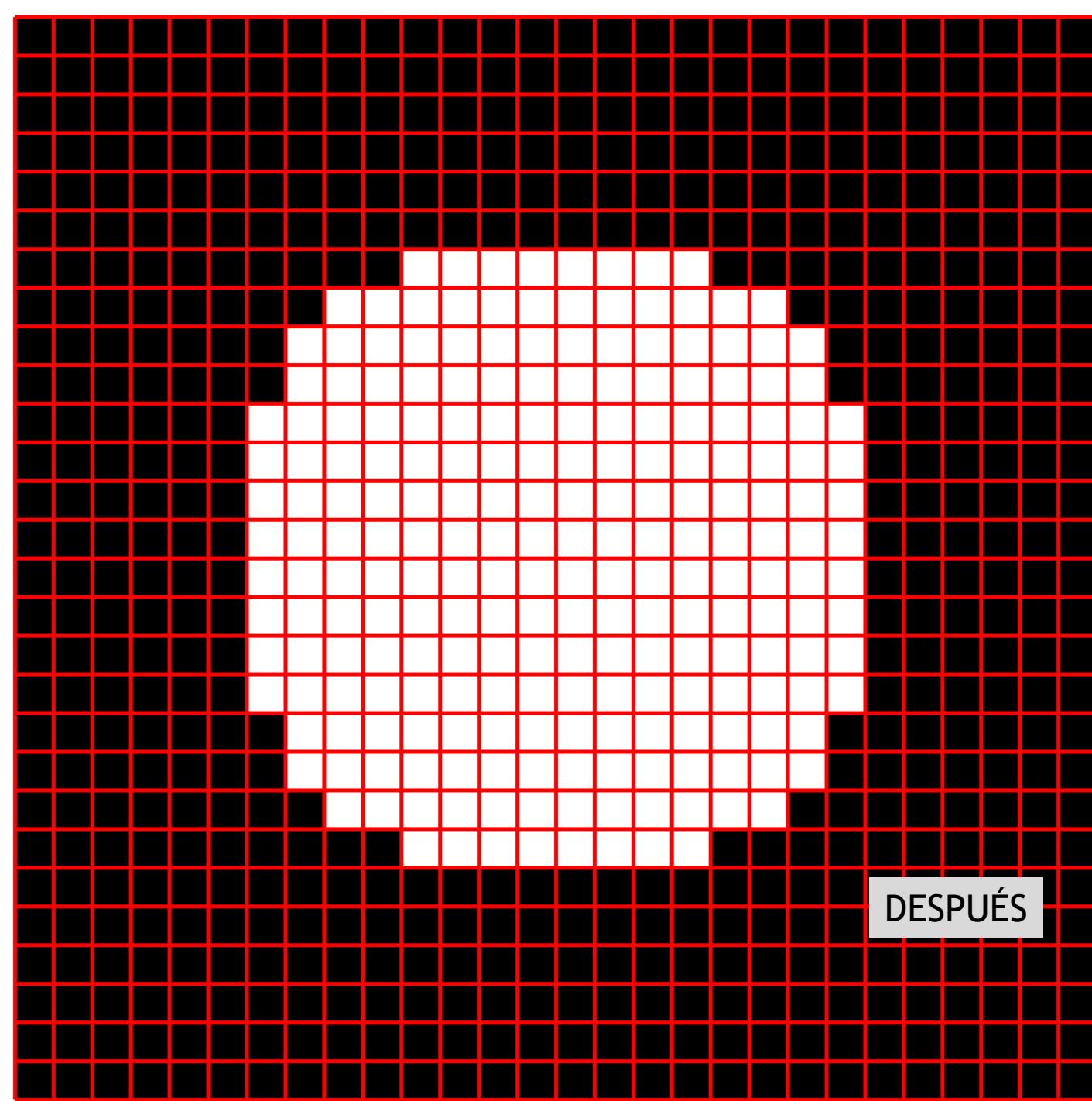


# Ejemplo Imagen binaria $28 \times 28$

Se desea  
eliminar  
rectángulo.

Erosión 5x5.  
Dilatación 5x5.

DESPUÉS

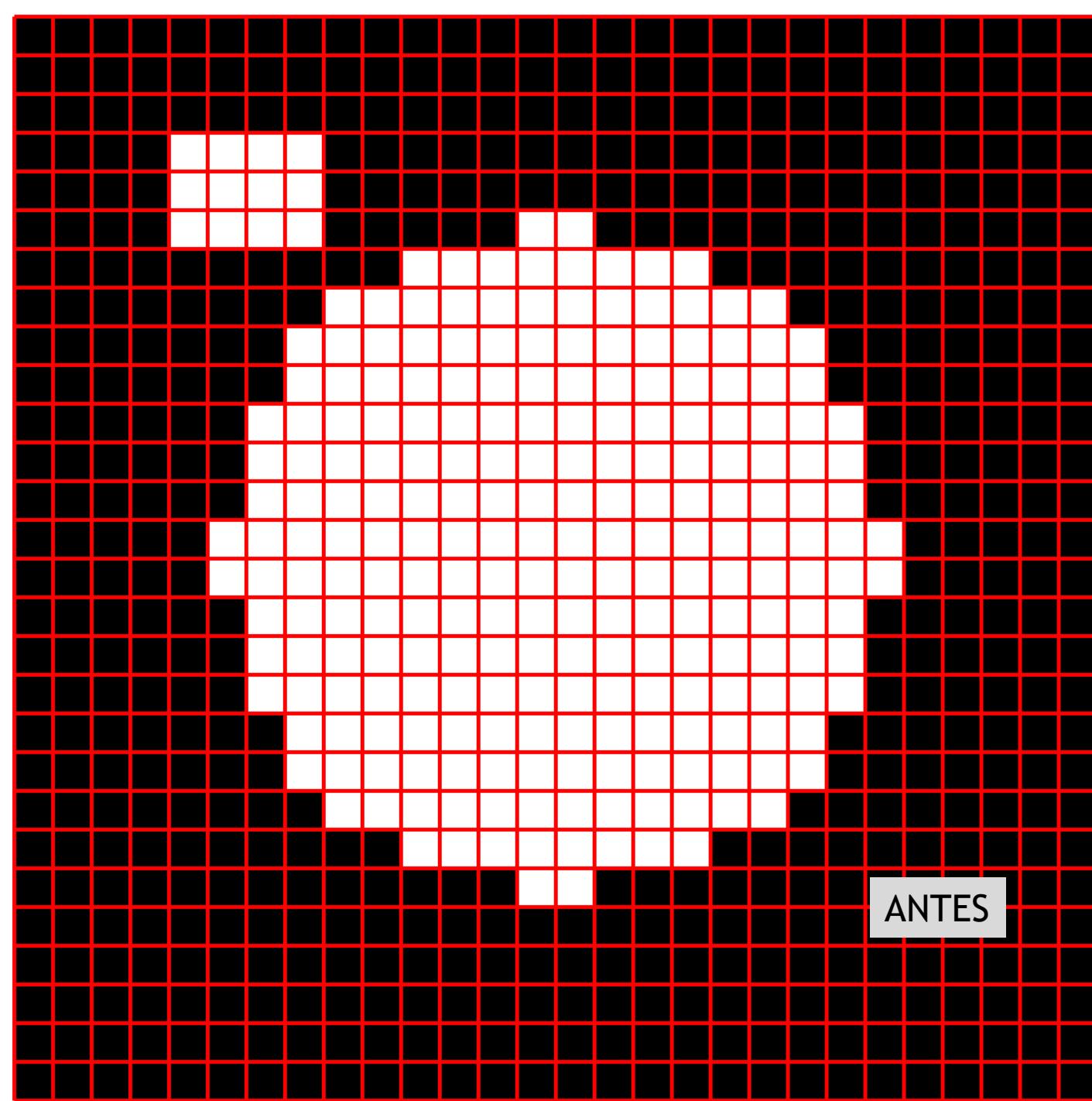


# Ejemplo Imagen binaria $28 \times 28$

Se desea  
eliminar  
rectángulo.

Erosión 5x5.  
Dilatación 5x5.

ANTES



# Apertura

$$\mathbf{Y} = \mathbf{X} \circ \mathbf{S} = (\mathbf{X} \ominus \mathbf{S}) \oplus \mathbf{S}$$

|      |      |  
Output   Input   Estructura

1. Erosión
2. Dilatación

# Dilatación y Erosión

$$Y = X \oplus S$$

- Dilatación: El output Y es ‘1’ si hay coincidencia en al menos un elemento de la estructura S con los elementos que cubre en X.
  - Esta operación equivale al máximo.

$$Y = X \ominus S$$

- Erosión: El output es ‘1’ sólo si hay coincidencia de todos los elementos de la estructura S con los elementos que cubre en X.
  - Esta operación equivale al mínimo.

# Dilatación y Erosión

En imágenes en tonos de gris (no binarias)

$$\mathbf{Y} = \mathbf{X} \oplus \mathbf{S}$$

- Esta operación equivale al máximo.

$$\mathbf{Y} = \mathbf{X} \ominus \mathbf{S}$$

- Esta operación equivale al mínimo.

Original  
1166 x 1556





Dilatación  
3x3



Dilatación  
5x5



Dilatación  
15x15

Original  
1166 x 1556





Erosión  
3x3



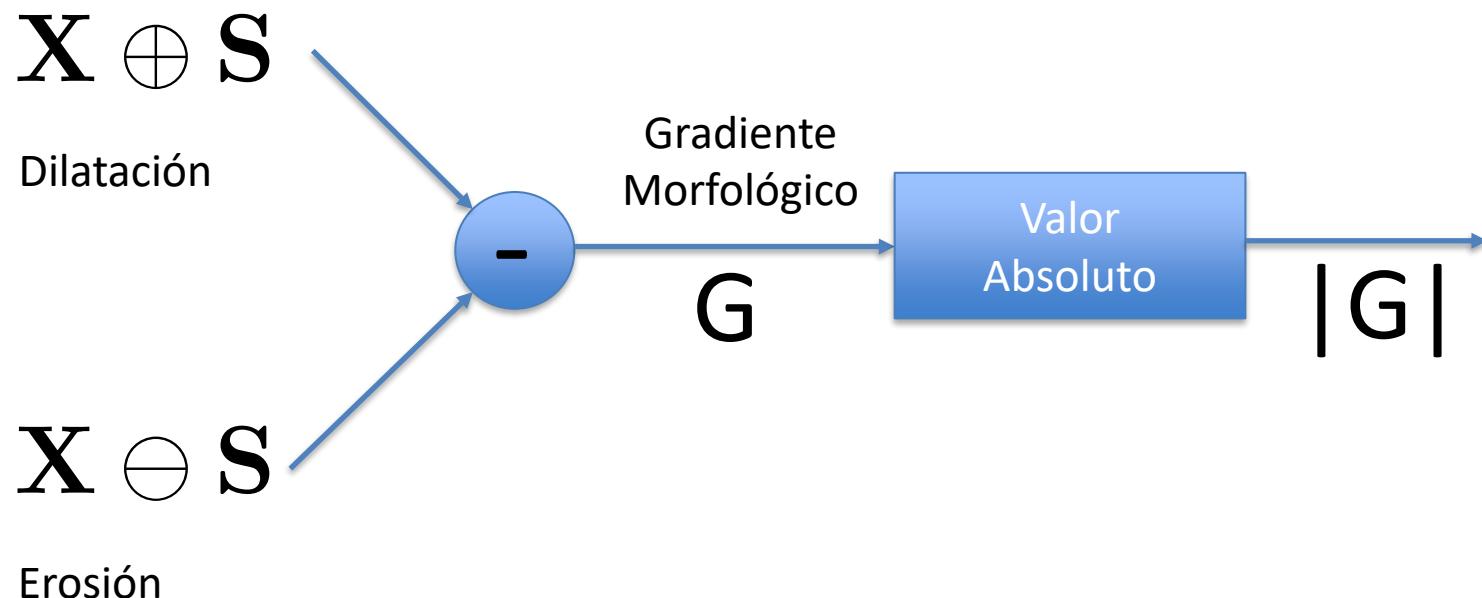
Erosión  
5x5



Erosión  
15x15

# Gradiente Morfológico

En imágenes en tonos de gris (no binarias)



A black and white photograph showing a close-up view of a weathered wooden building's exterior. The building features several arched windows with decorative wooden frames. The wood is dark and shows significant signs of age and wear. The sky in the background is overcast.

Original  
1166 x 1556

Gradiente  
3x3



Gradiente  
5x5



Gradiente  
15x15



Otro ejemplo

