# TRATAMIENTO DE SOMBRAS EN FOTOGRAFIAS DE DOCUMENTOS

```
# importamos librerias utilizadas
import cv2
import numpy as np
import matplotlib.pyplot as plt
from scipy import ndimage, misc
import matplotlib.pyplot as plt

# definimos unafuncion para graficar las imagenes
def imprimir_pic(pic,etiqueta):
    plt.figure()
    plt.imshow(pic, cmap = 'gray')
    plt.title(etiqueta)
    plt.axis('off')
```

### Seleccionamos la imagen con la cual se va a trabajar

```
imagen_entrada = "input8"
input_pic = cv2.imread(imagen_entrada+".jpeg")
```

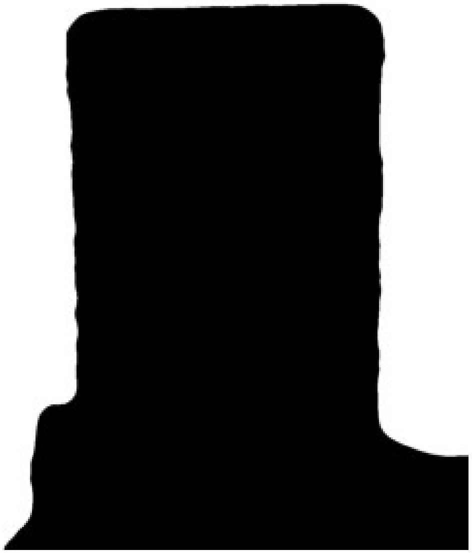
# METODO BASE

*Generamos la mascara o "perfil de sombra"*

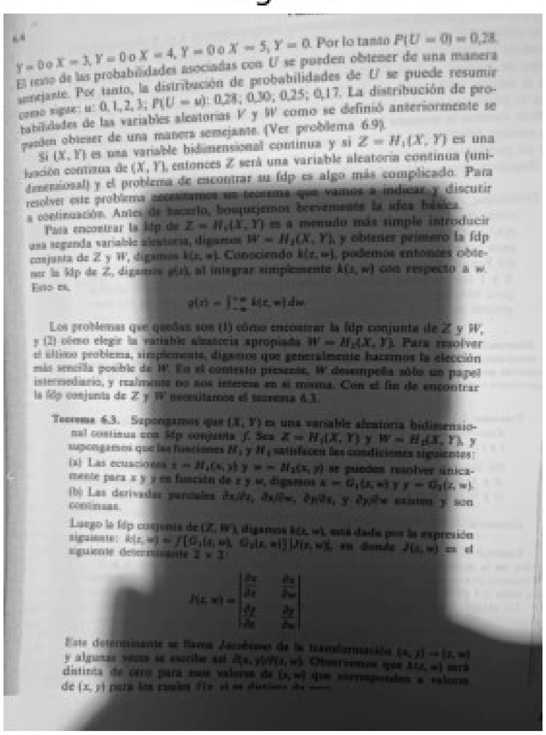
```
# Paquetes necesarios para la morfología matemática
from skimage.morphology import erosion, dilation, opening, closing
# Elementos estructurales
from skimage.morphology import disk, diamond, ball, rectangle, star
from scipy import ndimage as ndi
#reducimos la pic a grises
input_pic = cv2.cvtColor(input_pic, cv2.COLOR_BGR2GRAY)

#realizamos una copia de la pic recibida para poder modificarla en un
espacio de memoria diferente
mask = np.copy(input_pic)
#las sisguientes 3 lineas se encargar de normalizar la pic de 0 a 255
mask = mask - mask.min()
mask = mask/mask.max()
mask = mask*255


#Creamos la mascara con un filtro de mediana, consideramos las letras
como ruido

for i in range(3):
```

```
    mask = ndimage.median_filter(mask, size=18,mode='reflect')

mask_gris = np.copy(mask)


imprimir_pic(mask,'Máscara en grises')




#la maskara debe estar en blanco y negro y no en escala de grises, asi
que se realiza un if:
promedio = np.mean(mask)
for (cor_y,cor_x), value in np.ndenumerate(mask):
    if mask[cor_y,cor_x]<promedio: #Good Pixel
        mask[cor_y,cor_x] = 0.1
    elif mask[cor_y,cor_x]>promedio: #Bad Pixel
        mask[cor_y,cor_x] = 254.9


imprimir_pic(mask,'Máscara')
imprimir_pic(input_pic,'original')
```

Máscara



original



$Y = 0$ o $X = 3, Y = 0$ o $X = 4, Y = 0$ o $X = 5, Y = 0$. Por lo tanto $P(U = 0) = 0{,}28$. El resto de las probabilidades asociadas con $U$ se pueden obtener de una manera semejante. Por tanto, la distribución de probabilidades de $U$ se puede resumir como sigue: $u$: 0, 1, 2, 3; $P(U = u)$: 0,28; 0,30; 0,25; 0,17. La distribución de probabilidades de las variables aleatorias $V$ y $W$ como se definió anteriormente se pueden obtener de una manera semejante. (Ver problema 6.9).

Si $(X, Y)$ es una variable bidimensional continua y si $Z = H_1(X, Y)$ es una función continua de $(X, Y)$, entonces $Z$ será una variable aleatoria continua (unidimensional) y el problema de encontrar su fdp es algo más complicado. Para resolver este problema necesitamos un teorema que vamos a indicar y discutir a continuación. Antes de hacerlo, bosquejemos brevemente la idea básica.

Para encontrar la fdp de $Z = H_1(X, Y)$ es a menudo más simple introducir una segunda variable aleatoria, digamos $W = H_2(X, Y)$, y obtener primero la fdp conjunta de $Z$ y $W$, digamos $k(z, w)$. Conociendo $k(z, w)$, podemos entonces obtener la fdp de $Z$, digamos $g(z)$, al integrar simplemente $k(z, w)$ con respecto a $w$. Esto es,

$$g(z) = \int_{-\infty}^{+\infty} k(z, w)\, dw.$$

Los problemas que quedan son (1) cómo encontrar la fdp conjunta de $Z$ y $W$, y (2) cómo elegir la variable aleatoria apropiada $W = H_2(X, Y)$. Para resolver el último problema, simplemente, digamos que generalmente hacemos la elección más sencilla posible de $W$. En el contexto presente, $W$ desempeña sólo un papel intermediario, y realmente no nos interesa en sí misma. Con el fin de encontrar la fdp conjunta de $Z$ y $W$ necesitamos el teorema 6.3.

**Teorema 6.3.** Supongamos que $(X, Y)$ es una variable aleatoria bidimensional continua con fdp conjunta $f$. Sea $Z = H_1(X, Y)$ y $W = H_2(X, Y)$, y supongamos que las funciones $H_1$ y $H_2$ satisfacen las condiciones siguientes:

(a) Las ecuaciones $z = H_1(x, y)$ y $w = H_2(x, y)$ se pueden resolver únicamente para $x$ y $y$ en función de $z$ y $w$, digamos $x = G_1(z, w)$ y $y = G_2(z, w)$.

(b) Las derivadas parciales $\partial x/\partial z$, $\partial x/\partial w$, $\partial y/\partial z$, y $\partial y/\partial w$ existen y son continuas.

Luego la fdp conjunta de $(Z, W)$, digamos $k(z, w)$, está dada por la expresión siguiente: $k(z, w) = f[G_1(z, w), G_2(z, w)]\,|J(z, w)|$, en donde $J(z, w)$ es el siguiente determinante $2 \times 2$:

$$J(z, w) = \begin{vmatrix} \dfrac{\partial x}{\partial z} & \dfrac{\partial x}{\partial w} \\ \dfrac{\partial y}{\partial z} & \dfrac{\partial y}{\partial w} \end{vmatrix}$$

Este determinante se llama *Jacobiano* de la transformación $(x, y) \to (z, w)$ y algunas veces se escribe así $\partial(x, y)/\partial(z, w)$. Observemos que $k(z, w)$ será distinta de cero para esos valores de $(z, w)$ que corresponden a valores de $(x, y)$ para los cuales $f(x, y)$ es distinta de cero.

**Definimos la funcion de contraste que apartir de un percentil permite**

```
def funcion_de_contraste(pic,percentil):
    limite = np.percentile(pic,percentil)
    blancos = pic >= limite
    pic = pic + 255*blancos
    negros_saturados = pic<limite
    blancos_saturados = pic>=limite
    pic[negros_saturados] = 0
    pic[blancos_saturados] = 255
    return pic
```

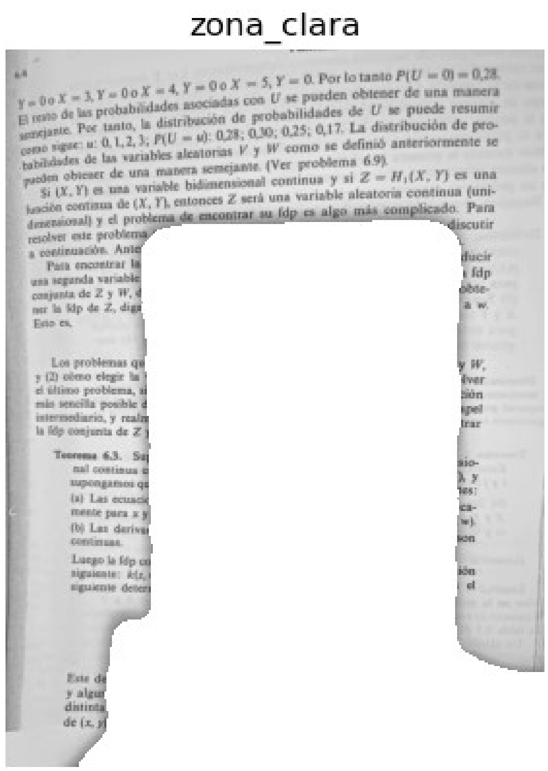
**Definimos el array "Zona Oscura"**

```
zona_oscura = np.full(input_pic.shape,np.nan)
limite = np.nanmean(input_pic)
for (coor_y,coor_x), value in np.ndenumerate(input_pic):
    aux = mask[coor_y:coor_y+1,coor_x:coor_x+1]
    if aux<limite: #Good Pixel
        zona_oscura[coor_y,coor_x] =  input_pic[coor_y,coor_x]
    elif aux>limite: #Bad Pixel
        None
imprimir_pic(zona_oscura,"zona_oscura")
```


**Definimos el array "Zona Clara"**

```
zona_clara = np.full(input_pic.shape,np.nan)
limite = np.nanmean(input_pic)
```

```python
for (coor_y,coor_x), value in np.ndenumerate(input_pic):
    aux = mask[coor_y:coor_y+1,coor_x:coor_x+1]
    if aux>limite: #Good Pixel
        zona_clara[coor_y,coor_x] =  input_pic[coor_y,coor_x]
    elif aux<=limite: #Bad Pixel
        None

imprimir_pic(zona_clara,"zona_clara")
```




**Definimos la funcion que nos permite conocer el punto optimo para contrastar la imagen**

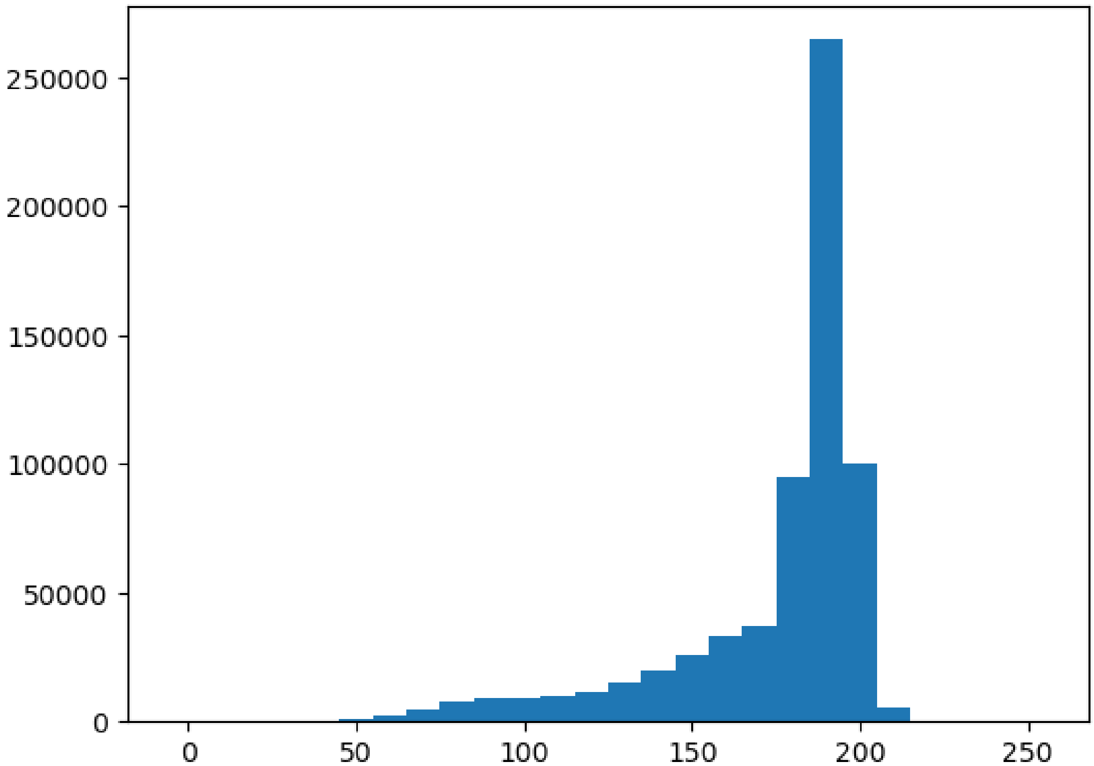
```python
def contraste_optimo_por_MSE(segmento,pic):

    output = 255 - np.nan_to_num(segmento,nan=0)

    base = 255 -pic
    errorlist = []
    for i in range(30):

        f = funcion_de_contraste(output,i)
        error = np.sum((base-f)**2)
        errorlist.append(error)

    m = np.arange(30)
    #plt.scatter( m,errorlist)
    #plt.title("Error vs intensidad (por percentile)")
```

```
    #plt.show()
    return (m[np.argmin(errorlist)]+1)
```

## METODO BASE

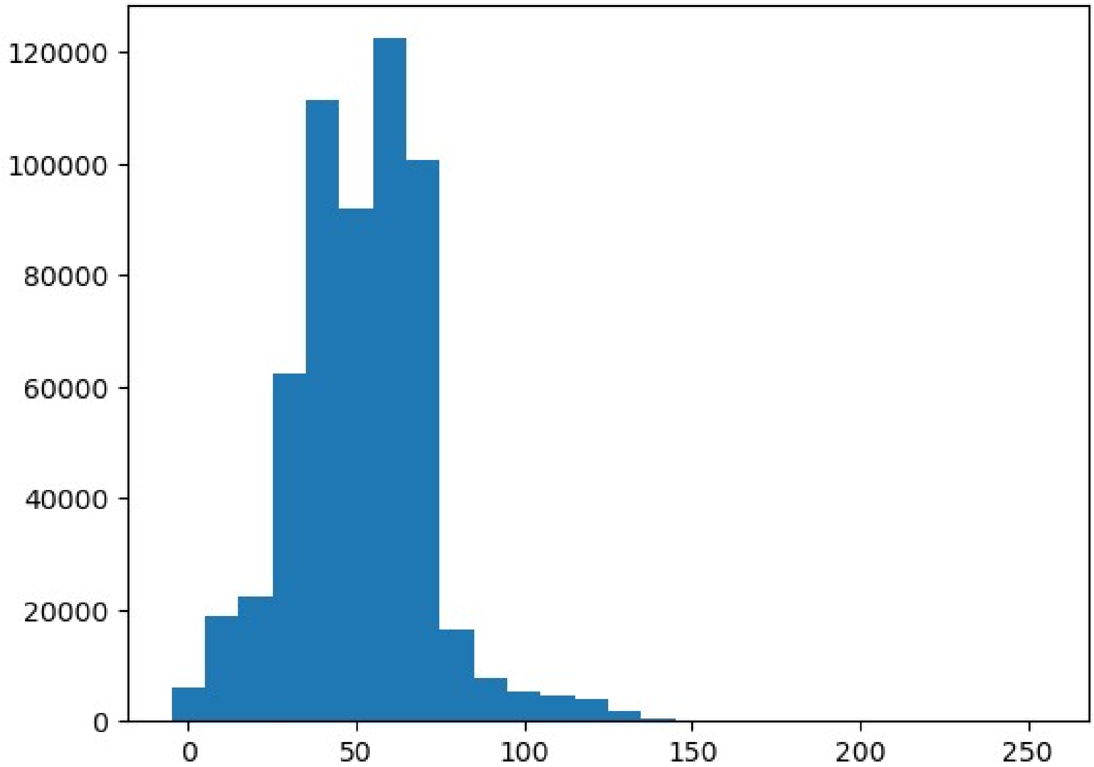
*Obtenemos el histograma de la zona clara con el fin de obtener su moda*

```
a,b = np.histogram(zona_clara,bins=np.arange(0,270,10))
plt.bar( b[0:26],a[0:26],width=10)
moda_zona_clara = b[np.argmax(a)+1]
```




*Obtenemos el histograma de la zona oscura con el fin de obtener su moda*

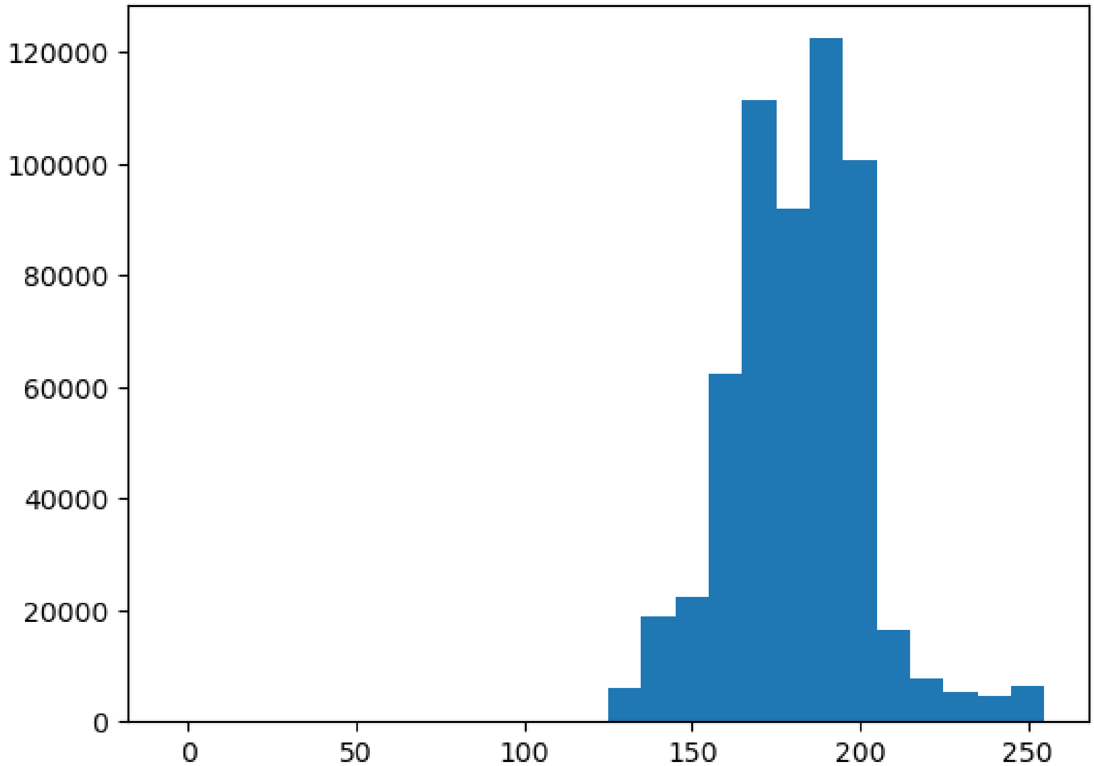
```
a,b = np.histogram(zona_oscura,bins=np.arange(0,270,10))
plt.bar( b[0:26],a[0:26],width=10)
moda_zona_oscura = b[np.argmax(a)+1]
```




*Modificamos el histograma de la zona oscura con el fin de igualar la moda de ambas zonas*

```
osc = np.copy(zona_oscura) + ( -moda_zona_oscura + moda_zona_clara)
#osc = zona_oscura
lessThen0 = osc<0
moreThen255 = osc>255
osc[lessThen0] = 0
osc[moreThen255] = 255
#imprimir_pic(osc,"osc")
a,b = np.histogram(osc,bins=np.arange(0,270,10))
plt.bar( b[0:26],a[0:26],width=10)
#moda_zona_oscura = b[np.argmax(a)+1]
```

```
<BarContainer object of 26 artists>
```




```
imprimir_pic(osc,"Zona oscura con histograma modificado")
```

Zona oscura con histograma modificado


de hacerlo, bosquejemos brevemente la idea básic
fdp de $Z = H_1(X, Y)$ es a menudo más simple intr
aleatoria, digamos $W = H_2(X, Y)$, y obtener primero l
gamos $k(z, w)$. Conociendo $k(z, w)$, podemos entonces
nos $g(z)$, al integrar simplemente $k(z, w)$ con respecto

$$g(z) = \int_{-\infty}^{+\infty} k(z, w)\, dw.$$

quedan son (1) cómo encontrar la fdp conjunta de $Z$
ariable aleatoria apropiada $W = H_2(X, Y)$. Para res
mplemente, digamos que generalmente hacemos la elec
$W$. En el contexto presente, $W$ desempeña sólo un p
ente no nos interesa en sí misma. Con el fin de enco
$W$ necesitamos el teorema 6.3.

ongamos que $(X, Y)$ es una variable aleatoria bidime
n fdp conjunta $f$. Sea $Z = H_1(X, Y)$ y $W = H_2(X,$
las funciones $H_1$ y $H_2$ satisfacen las condiciones siguie
nes $z = H_1(x, y)$ y $w = H_2(x, y)$ se pueden resolver ún
en función de $z$ y $w$, digamos $x = G_1(z, w)$ y $y = G_2($
as parciales $\partial x/\partial z$, $\partial x/\partial w$, $\partial y/\partial z$, y $\partial y/\partial w$ existen y

junta de $(Z, W)$, digamos $k(z, w)$, está dada por la expre
$) = f[G_1(z, w), G_2(z, w)]\,|J(z, w)|$, en donde $J(z, w)$ e
inante 2 × 2:

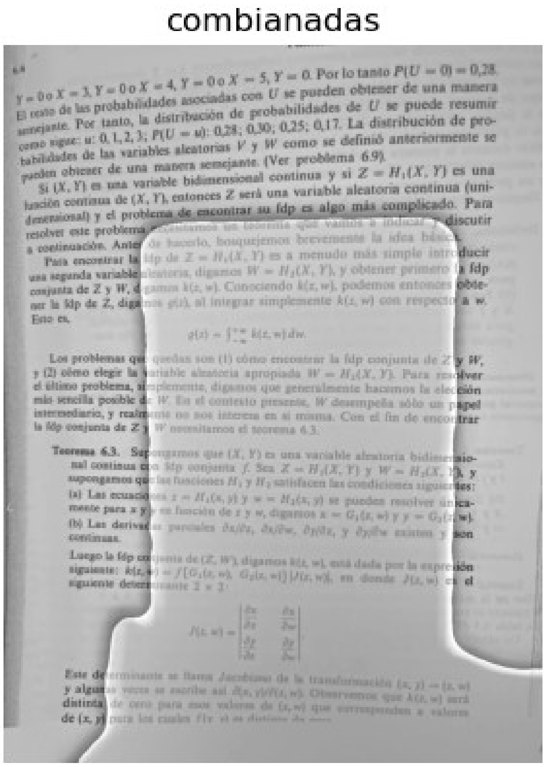
$$J(z, w) = \begin{vmatrix} \frac{\partial x}{\partial z} & \frac{\partial x}{\partial w} \\ \frac{\partial y}{\partial z} & \frac{\partial y}{\partial w} \end{vmatrix}.$$

erminante se llama *Jacobiano* de la transformación $(x, y) \to (z, w)$
as veces se escribe así $\partial(x, y)/\partial(z, w)$. Observemos que $k(z, w)$ será
de cero para esos valores de $(z, w)$ que corresponden a valores
para los cuales $f(x, y)$ es distinta de cero.

```
imprimir_pic(zona_clara,"zona_clara")
```


*Combinamos ambas zonas: la zona clara y la zona oscura modificada*

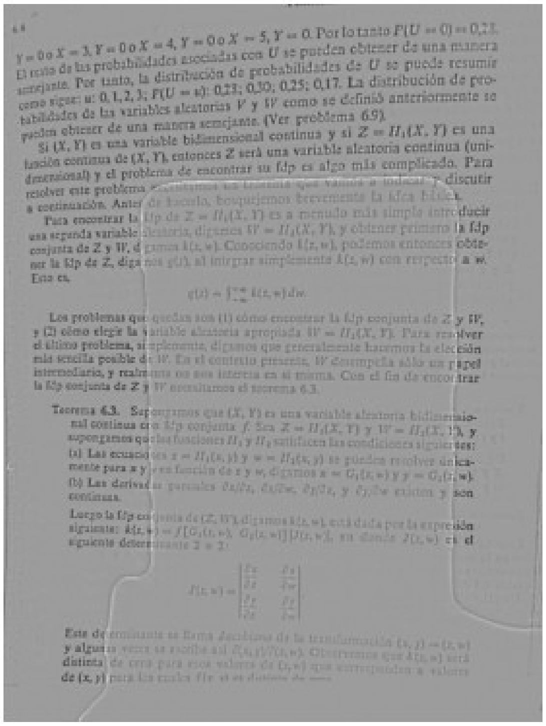
```
z1 =  np.nan_to_num(zona_clara,0)
z2 = np.nan_to_num(osc,0)

suma_parcial = (z1 + z2)
imprimir_pic(suma_parcial,"combianadas")
```




*Restamos las mascara (en tonos de gris) de la imagen original*

```
resta_parcial = np.copy(suma_parcial)

mask_gris = ndimage.median_filter(suma_parcial,
size=15,mode='reflect')


resta_parcial = ((suma_parcial - mask_gris)+255)/(2)
lessThen0 = resta_parcial<0
moreThen255 = resta_parcial>255
resta_parcial[lessThen0] = 0
resta_parcial[moreThen255] = 255


imprimir_pic(resta_parcial,"resta")
```

resta


$Y = 0$ o $X = 3, Y = 0$ o $X = 4, Y = 0$ o $X = 5, Y = 0$. Por lo tanto $P(U = 0) = 0{,}23$. El resto de las probabilidades asociadas con $U$ se pueden obtener de una manera semejante. Por tanto, la distribución de probabilidades de $U$ se puede resumir como sigue: u: 0, 1, 2, 3; $P(U = u)$: 0,23; 0,30; 0,25; 0,17. La distribución de probabilidades de las variables aleatorias $V$ y $W$ como se definió anteriormente se pueden obtener de una manera semejante. (Ver problema 6.9).

Si $(X, Y)$ es una variable bidimensional continua y si $Z = H_1(X, Y)$ es una función continua de $(X, Y)$, entonces $Z$ será una variable aleatoria continua (unidimensional) y el problema de encontrar su fdp es algo más complicado. Para resolver este problema [illegible] discutir a continuación. Antes de hacerlo, bosquejemos brevemente la idea básica.

Para encontrar la fdp de $Z = H_1(X, Y)$ es a menudo más simple introducir una segunda variable aleatoria, digamos $W = H_2(X, Y)$, y obtener primero la fdp conjunta de $Z$ y $W$, digamos $k(z, w)$. Conociendo $k(z, w)$, podemos entonces obtener la fdp de $Z$, digamos $g(z)$, al integrar simplemente $k(z, w)$ con respecto a $w$. Esto es,

$$g(z) = \int_{-\infty}^{+\infty} k(z, w)\, dw.$$

Los problemas que quedan son (1) cómo encontrar la fdp conjunta de $Z$ y $W$, y (2) cómo elegir la variable aleatoria apropiada $W = H_2(X, Y)$. Para resolver el último problema, simplemente, digamos que generalmente hacemos la elección más sencilla posible de $W$. En el contexto presente, $W$ desempeña sólo un papel intermediario, y realmente no nos interesa en sí misma. Con el fin de encontrar la fdp conjunta de $Z$ y $W$ necesitamos el teorema 6.3.

**Teorema 6.3.** Supongamos que $(X, Y)$ es una variable aleatoria bidimensional continua con fdp conjunta $f$. Sea $Z = H_1(X, Y)$ y $W = H_2(X, Y)$, y supongamos que las funciones $H_1$ y $H_2$ satisfacen las condiciones siguientes:
(a) Las ecuaciones $z = H_1(x, y)$ y $w = H_2(x, y)$ se pueden resolver únicamente para $x$ y $y$ en función de $z$ y $w$, digamos $x = G_1(z, w)$ y $y = G_2(z, w)$.
(b) Las derivadas parciales $\partial x/\partial z$, $\partial x/\partial w$, $\partial y/\partial z$, y $\partial y/\partial w$ existen y son continuas.

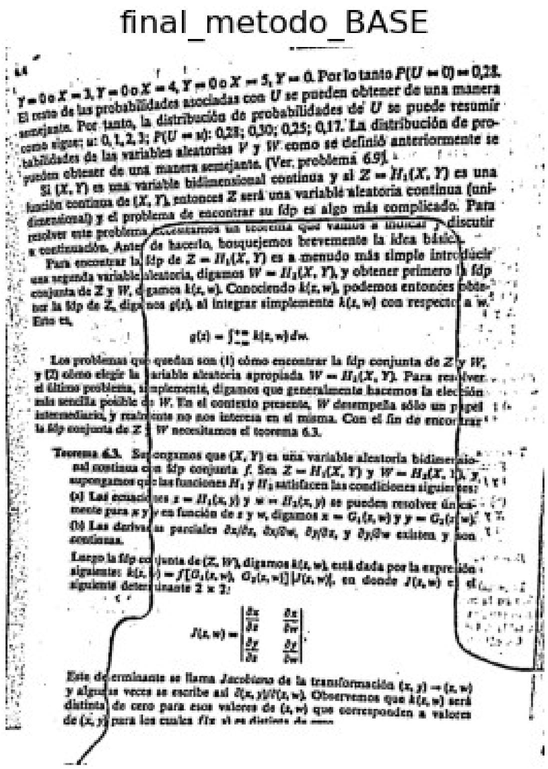
Luego la fdp conjunta de $(Z, W)$, digamos $k(z, w)$, está dada por la expresión siguiente: $k(z, w) = f[G_1(z, w), G_2(z, w)]\,|J(z, w)|$, en donde $J(z, w)$ es el siguiente determinante $2 \times 2$:

$$J(z, w) = \begin{vmatrix} \frac{\partial x}{\partial z} & \frac{\partial x}{\partial w} \\ \frac{\partial y}{\partial z} & \frac{\partial y}{\partial w} \end{vmatrix}$$

Este determinante se llama *Jacobiano* de la transformación $(x, y) \to (z, w)$ y algunas veces se escribe así $\partial(x, y)/\partial(z, w)$. Observemos que $k(z, w)$ será distinta de cero para esos valores de $(z, w)$ que corresponden a valores de $(x, y)$ para los cuales $f(x, y)$ es distinta de cero.

*Aplicamos un contraste optmizado por MSE a la imagen de "resta parcial"*

```
p_indicado = contraste_optimo_por_MSE(255 - resta_parcial,255 -
input_pic)
final_metodo_BASE = funcion_de_contraste(resta_parcial,p_indicado)
imprimir_pic(final_metodo_BASE,"final_metodo_BASE")
```




Y = 0 o X = 3, Y = 0 o X = 4, Y = 0 o X = 5, Y = 0. Por lo tanto P(U = 0) = 0,28. El resto de las probabilidades asociadas con U se pueden obtener de una manera semejante. Por tanto, la distribución de probabilidades de U se puede resumir como sigue: u: 0, 1, 2, 3; P(U = u): 0,28; 0,30; 0,25; 0,17. La distribución de probabilidades de las variables aleatorias V y W como se definió anteriormente se pueden obtener de una manera semejante. (Ver. problema 6.9.)

Si (X, Y) es una variable bidimensional continua y si Z = H₁(X, Y) es una función continua de (X, Y), entonces Z será una variable aleatoria continua (unidimensional) y el problema de encontrar su fdp es algo más complicado. Para resolver este problema ... a continuación. Antes de hacerlo, bosquejemos brevemente la idea básica.

Para encontrar la fdp de Z = H₁(X, Y) es a menudo más simple introducir una segunda variable aleatoria, digamos W = H₂(X, Y), y obtener primero la fdp conjunta de Z y W, digamos k(z, w). Conociendo k(z, w), podemos entonces obtener la fdp de Z, digamos g(z), al integrar simplemente k(z, w) con respecto a w. Esto es,

$$g(z) = \int_{-\infty}^{+\infty} k(z, w)\, dw.$$

Los problemas que quedan son (1) cómo encontrar la fdp conjunta de Z y W, y (2) cómo elegir la variable aleatoria apropiada W = H₂(X, Y). Para resolver el último problema, simplemente, digamos que generalmente hacemos la elección más sencilla posible de W. En el contexto presente, W desempeña sólo un papel intermediario, y realmente no nos interesa en sí misma. Con el fin de encontrar la fdp conjunta de Z y W necesitamos el teorema 6.3.

**Teorema 6.3.** Supongamos que (X, Y) es una variable aleatoria bidimensional continua con fdp conjunta f. Sea Z = H₁(X, Y) y W = H₂(X, Y), y supongamos que las funciones H₁ y H₂ satisfacen las condiciones siguientes: (a) Las ecuaciones z = H₁(x, y) y w = H₂(x, y) se pueden resolver únicamente para x y y en función de z y w, digamos x = G₁(z, w) y y = G₂(z, w). (b) Las derivadas parciales ∂x/∂z, ∂x/∂w, ∂y/∂z, y ∂y/∂w existen y son continuas.

Luego la fdp conjunta de (Z, W), digamos k(z, w), está dada por la expresión siguiente: k(z, w) = f[G₁(z, w), G₂(z, w)] |J(z, w)|, en donde J(z, w) es el siguiente determinante 2 × 2:

$$J(z, w) = \begin{vmatrix} \frac{\partial x}{\partial z} & \frac{\partial x}{\partial w} \\ \frac{\partial y}{\partial z} & \frac{\partial y}{\partial w} \end{vmatrix}.$$

Este determinante se llama *Jacobiano* de la transformación (x, y) → (z, w) y algunas veces se escribe así ∂(x, y)/∂(z, w). Observemos que k(z, w) será distinta de cero para esos valores de (z, w) que corresponden a valores de (x, y) para los cuales f(x, y) es distinta de ...

## METODO B

```
# Importar las librearias necesarias:

import matplotlib.pyplot as plt
from PIL import Image
import scipy.ndimage as snd
import numpy as np
import cv2
from skimage.filters import threshold_otsu
from cv2 import threshold, adaptiveThreshold

# Declarar metodos para aplicar Clausura y Threshold:

def closing(image_array):

  # Estructurar elemento de clausura:
  huella = np.ones((40, 40))
  # Aplicar clausura de grises:
  fondo = snd.grey_closing(image_array, footprint=huella)
  # Se resta el fondo de la imagen:
  fondo_libre = (image_array.astype(np.float64) -
fondo.astype(np.float64))
  # Se reescala la imagen con el fondo libre de 0 a 255:
  denominador = (fondo_libre.max() - fondo_libre.min())
```

```
  fondo_libre_normalizado = (fondo_libre - fondo_libre.min())*
255/denominador
  # Convertir fondo_libre_normalizado a uint8:
  fondo_libre_normalizado = fondo_libre_normalizado.astype(np.uint8)
  # Convertir fondo_libre_normalizado a imagen:
  fondo_libre_normalizado = Image.fromarray(fondo_libre_normalizado)
  return fondo_libre_normalizado


def threshold(image_array):
  th = cv2.adaptiveThreshold(np.asarray(image_array),
    255,  # Maximo valor assignado al valor de un pixel que excede el
'threshold'.
    cv2.ADAPTIVE_THRESH_MEAN_C,  # suma ponderada "gaussiana" de los
vecinos.
    cv2.THRESH_BINARY,  # Tipo de threshold.
    15,  # Tamaño del bloque (ventana 5x5)
    12)

  # Convertir th a imagen:
  th = Image.fromarray(th)

  return th

# Importar imagen:
input_image = input_pic
# Convertir la imagen en arreglo:
image_array = np.asarray(input_image)
# Aplicar Clausura:
closing_image = closing(image_array)
# Aplicar Threshold:
threshold_image = threshold(closing_image)
```

```
input_image
```

```
array([[146, 148, 151, ..., 179, 179, 178],
       [141, 143, 146, ..., 179, 178, 178],
       [139, 141, 144, ..., 178, 178, 177],
       ...,
       [190, 190, 190, ...,  36,  36,  36],
       [190, 190, 190, ...,  36,  36,  36],
       [190, 190, 190, ...,  36,  36,  36]], dtype=uint8)
```

```
closing_image
```

$Y = 0$ o $X = 3, Y = 0$ o $X = 4, Y = 0$ o $X = 5, Y = 0$. Por lo tanto $P(U = 0) = 0{,}28$. El resto de las probabilidades asociadas con $U$ se pueden obtener de una manera semejante. Por tanto, la distribución de probabilidades de $U$ se puede resumir como sigue: $u$: 0, 1, 2, 3; $P(U = u)$: 0,28; 0,30; 0,25; 0,17. La distribución de probabilidades de las variables aleatorias $V$ y $W$ como se definió anteriormente se pueden obtener de una manera semejante. (Ver problema 6.9).

Si $(X, Y)$ es una variable bidimensional continua y si $Z = H_1(X, Y)$ es una función continua de $(X, Y)$, entonces $Z$ será una variable aleatoria continua (unidimensional) y el problema de encontrar su fdp es algo más complicado. Para resolver este problema necesitamos un teorema que vamos a indicar y discutir a continuación. Antes de hacerlo, bosquejemos brevemente la idea básica.

Para encontrar la fdp de $Z = H_1(X, Y)$ es a menudo más simple introducir una segunda variable aleatoria, digamos $W = H_2(X, Y)$, y obtener primero la fdp conjunta de $Z$ y $W$, digamos $k(z, w)$. Conociendo $k(z, w)$, podemos entonces obtener la fdp de $Z$, digamos $g(z)$, al integrar simplemente $k(z, w)$ con respecto a $w$. Esto es,

$$g(z) = \int_{-\infty}^{+\infty} k(z, w)\, dw.$$

Los problemas que quedan son (1) cómo encontrar la fdp conjunta de $Z$ y $W$, y (2) cómo elegir la variable aleatoria apropiada $W = H_2(X, Y)$. Para resolver el último problema, simplemente, digamos que generalmente hacemos la elección más sencilla posible de $W$. En el contexto presente, $W$ desempeña sólo un papel intermediario, y realmente no nos interesa en sí misma. Con el fin de encontrar la fdp conjunta de $Z$ y $W$ necesitamos el teorema 6.3.

**Teorema 6.3.** Supongamos que $(X, Y)$ es una variable aleatoria bidimensional continua con fdp conjunta $f$. Sea $Z = H_1(X, Y)$ y $W = H_2(X, Y)$, y supongamos que las funciones $H_1$ y $H_2$ satisfacen las condiciones siguientes:

(a) Las ecuaciones $z = H_1(x, y)$ y $w = H_2(x, y)$ se pueden resolver únicamente para $x$ y $y$ en función de $z$ y $w$, digamos $x = G_1(z, w)$ y $y = G_2(z, w)$.

(b) Las derivadas parciales $\partial x/\partial z$, $\partial x/\partial w$, $\partial y/\partial z$, y $\partial y/\partial w$ existen y son continuas.

Luego la fdp conjunta de $(Z, W)$, digamos $k(z, w)$, está dada por la expresión siguiente: $k(z, w) = f[G_1(z, w),\ G_2(z, w)]\,|J(z, w)|$, en donde $J(z, w)$ es el siguiente determinante $2 \times 2$:

$$J(z, w) = \begin{vmatrix} \dfrac{\partial x}{\partial z} & \dfrac{\partial x}{\partial w} \\ \dfrac{\partial y}{\partial z} & \dfrac{\partial y}{\partial w} \end{vmatrix}.$$

Este determinante se llama *Jacobiano* de la transformación $(x, y) \to (z, w)$ y algunas veces se escribe así $\partial(x, y)/\partial(z, w)$. Observemos que $k(z, w)$ será distinta de cero para esos valores de $(z, w)$ que corresponden a valores de $(x, y)$ para los cuales $f(x, y)$ es distinta de cero.

$Y = 0$ o $X = 3, Y = 0$ o $X = 4, Y = 0$ o $X = 5, Y = 0$. Por lo tanto $P(U = 0) = 0{,}28$. El resto de las probabilidades asociadas con $U$ se pueden obtener de una manera semejante. Por tanto, la distribución de probabilidades de $U$ se puede resumir como sigue: $u$: 0, 1, 2, 3; $P(U = u)$: 0,28; 0,30; 0,25; 0,17. La distribución de probabilidades de las variables aleatorias $V$ y $W$ como se definió anteriormente se pueden obtener de una manera semejante. (Ver problema 6.9).

Si $(X, Y)$ es una variable bidimensional continua y si $Z = H_1(X, Y)$ es una función continua de $(X, Y)$, entonces $Z$ será una variable aleatoria continua (unidimensional) y el problema de encontrar su fdp es algo más complicado. Para resolver este problema necesitamos un teorema que vamos a indicar y discutir a continuación. Antes de hacerlo, bosquejemos brevemente la idea básica.

Para encontrar la fdp de $Z = H_1(X, Y)$ es a menudo más simple introducir una segunda variable aleatoria, digamos $W = H_2(X, Y)$, y obtener primero la fdp conjunta de $Z$ y $W$, digamos $k(z, w)$. Conociendo $k(z, w)$, podemos entonces obtener la fdp de $Z$, digamos $g(z)$, al integrar simplemente $k(z, w)$ con respecto a $w$. Esto es,

$$g(z) = \int_{-\infty}^{+\infty} k(z, w)\, dw.$$

Los problemas que quedan son (1) cómo encontrar la fdp conjunta de $Z$ y $W$, y (2) cómo elegir la variable aleatoria apropiada $W = H_2(X, Y)$. Para resolver el último problema, simplemente, digamos que generalmente hacemos la elección más sencilla posible de $W$. En el contexto presente, $W$ desempeña sólo un papel intermediario, y realmente no nos interesa en sí misma. Con el fin de encontrar la fdp conjunta de $Z$ y $W$ necesitamos el teorema 6.3.

**Teorema 6.3.** Supongamos que $(X, Y)$ es una variable aleatoria bidimensional continua con fdp conjunta $f$. Sea $Z = H_1(X, Y)$ y $W = H_2(X, Y)$, y supongamos que las funciones $H_1$ y $H_2$ satisfacen las condiciones siguientes:

(a) Las ecuaciones $z = H_1(x, y)$ y $w = H_2(x, y)$ se pueden resolver únicamente para $x$ y $y$ en función de $z$ y $w$, digamos $x = G_1(z, w)$ y $y = G_2(z, w)$.

(b) Las derivadas parciales $\partial x/\partial z$, $\partial x/\partial w$, $\partial y/\partial z$, y $\partial y/\partial w$ existen y son continuas.

Luego la fdp conjunta de $(Z, W)$, digamos $k(z, w)$, está dada por la expresión siguiente: $k(z, w) = f[G_1(z, w), G_2(z, w)]\,|J(z, w)|$, en donde $J(z, w)$ es el siguiente determinante $2 \times 2$:

$$J(z, w) = \begin{vmatrix} \dfrac{\partial x}{\partial z} & \dfrac{\partial x}{\partial w} \\ \dfrac{\partial y}{\partial z} & \dfrac{\partial y}{\partial w} \end{vmatrix}.$$

Este determinante se llama *Jacobiano* de la transformación $(x, y) \to (z, w)$ y algunas veces se escribe así $\partial(x, y)/\partial(z, w)$. Observemos que $k(z, w)$ será distinta de cero para esos valores de $(z, w)$ que corresponden a valores de $(x, y)$ para los cuales $f(x, y)$ es distinta de cero.

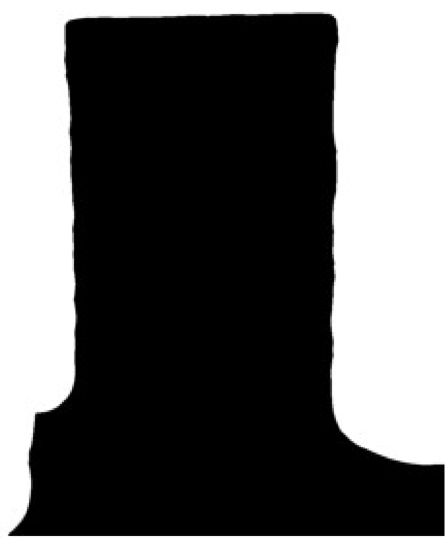
## METODO A

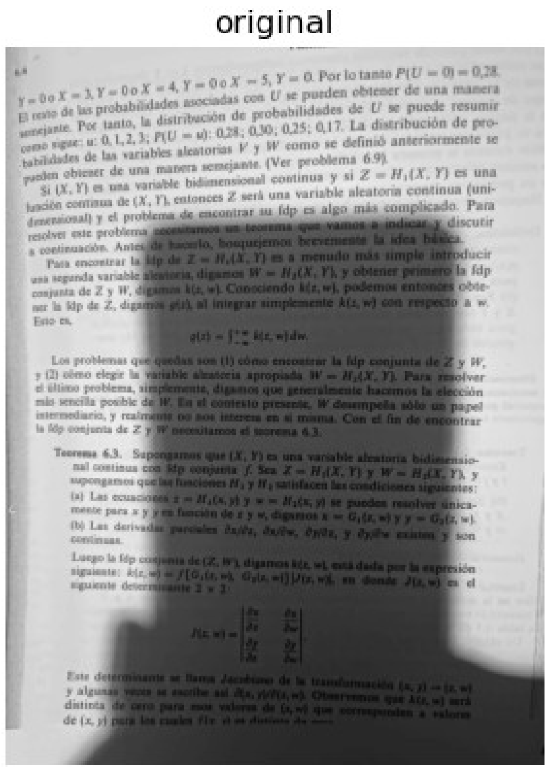
```
mask = dilation(image=mask, selem=disk(40))
```

```python
imprimir_pic(mask,'Máscara')
imprimir_pic(input_pic,'original')
```

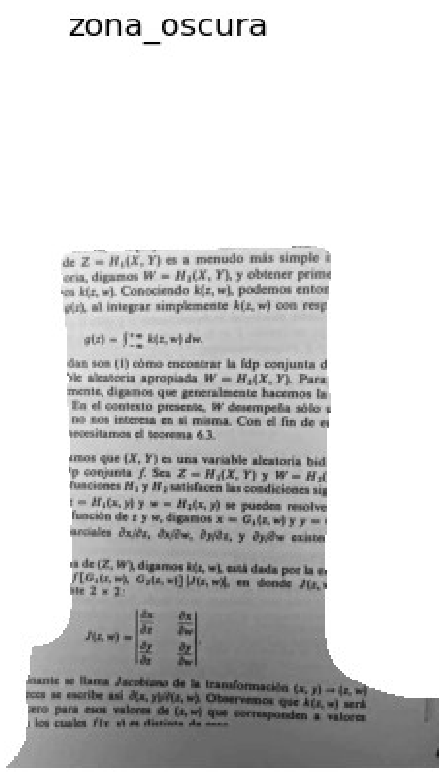
```
/tmp/ipykernel_12094/2346952556.py:1: FutureWarning: `selem` is a
deprecated argument name for `dilation`. It will be removed in version
1.0. Please use `footprint` instead.
  mask = dilation(image=mask, selem=disk(40))
```

Máscara

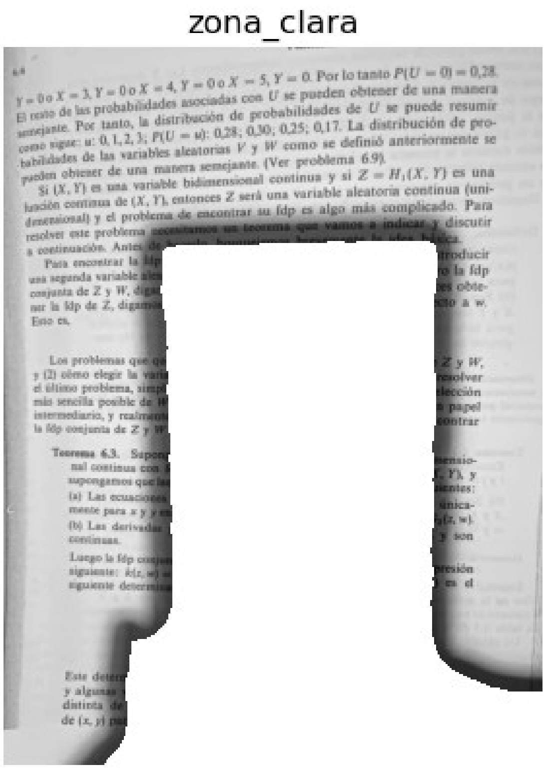




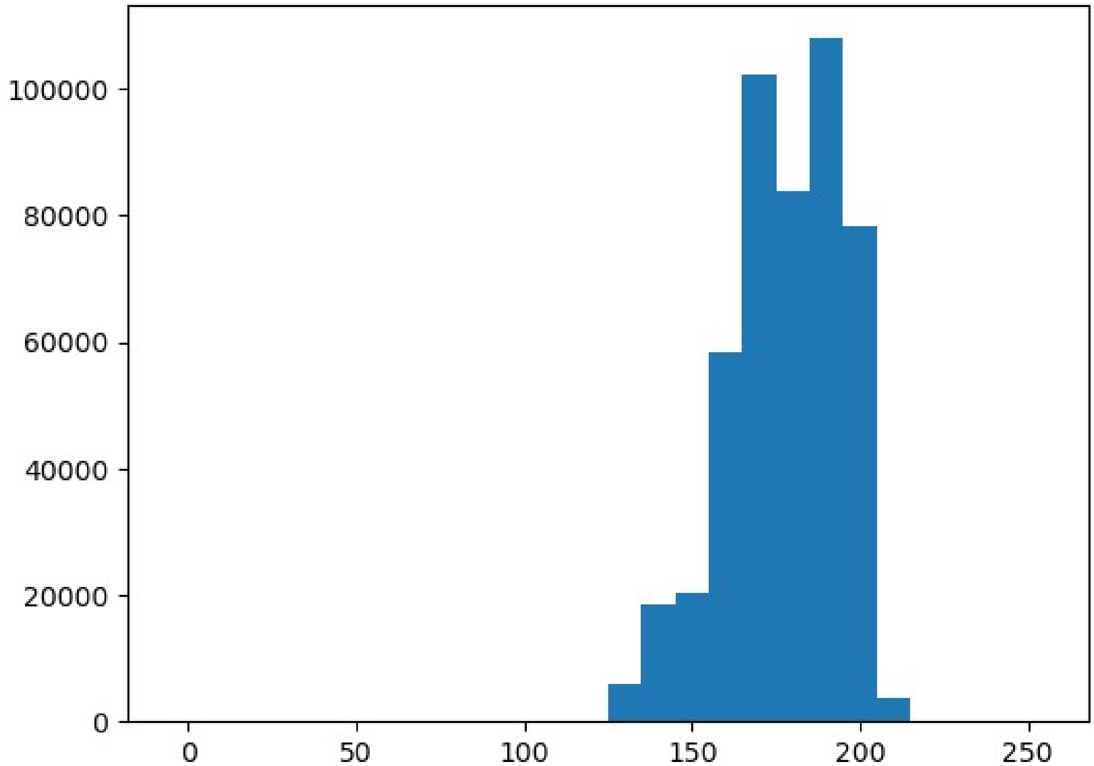

```python
zona_oscura = np.full(input_pic.shape,np.nan)
limite = np.nanmean(input_pic)
for (coor_y,coor_x), value in np.ndenumerate(input_pic):
    aux = mask[coor_y:coor_y+1,coor_x:coor_x+1]
    if aux<limite: #Good Pixel
        zona_oscura[coor_y,coor_x] =  input_pic[coor_y,coor_x]
    elif aux>limite: #Bad Pixel
        None
imprimir_pic(zona_oscura,"zona_oscura")
```



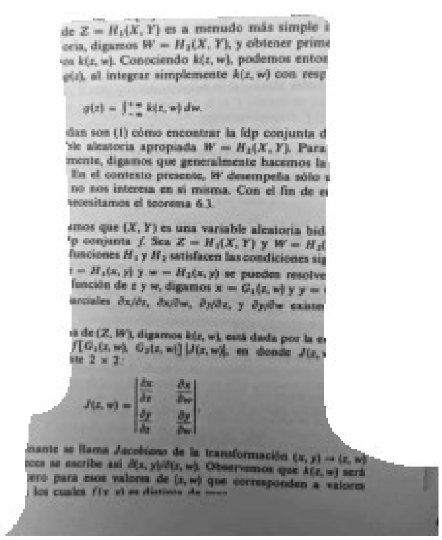

```python
zona_clara = np.full(input_pic.shape,np.nan)
limite = np.nanmean(input_pic)

for (coor_y,coor_x), value in np.ndenumerate(input_pic):
    aux = mask[coor_y:coor_y+1,coor_x:coor_x+1]
    if aux>limite: #Good Pixel
        zona_clara[coor_y,coor_x] =  input_pic[coor_y,coor_x]
    elif aux<=limite: #Bad Pixel
        None

imprimir_pic(zona_clara,"zona_clara")
```




```python
#a,b = np.histogram(zona_clara,bins=np.arange(0,270,10))
#plt.bar( b[0:26],a[0:26],width=10)
#moda_zona_clara = b[np.argmax(a)+1]
#
#a,b = np.histogram(zona_oscura,bins=np.arange(0,270,10))
#plt.bar( b[0:26],a[0:26],width=10)
#moda_zona_oscura = b[np.argmax(a)+1]

osc = np.copy(zona_oscura) + ( -moda_zona_oscura + moda_zona_clara)
#osc = zona_oscura
lessThen0 = osc<0
moreThen255 = osc>255
osc[lessThen0] = 0
osc[moreThen255] = 255
#imprimir_pic(osc,"osc")
a,b = np.histogram(osc,bins=np.arange(0,270,10))
plt.bar( b[0:26],a[0:26],width=10)
#moda_zona_oscura = b[np.argmax(a)+1]
imprimir_pic(osc,"Zona oscura con histograma modificado")
imprimir_pic(zona_clara,"zona_clara")
```




Zona oscura con histograma modificado



de $Z = H_1(X, Y)$ es a menudo más simple
oria, digamos $W = H_2(X, Y)$, y obtener prime
os $k(z, w)$. Conociendo $k(z, w)$, podemos enton
$g(z)$, al integrar simplemente $k(z, w)$ con resp

$$g(z) = \int_{-\infty}^{+\infty} k(z, w)\, dw.$$

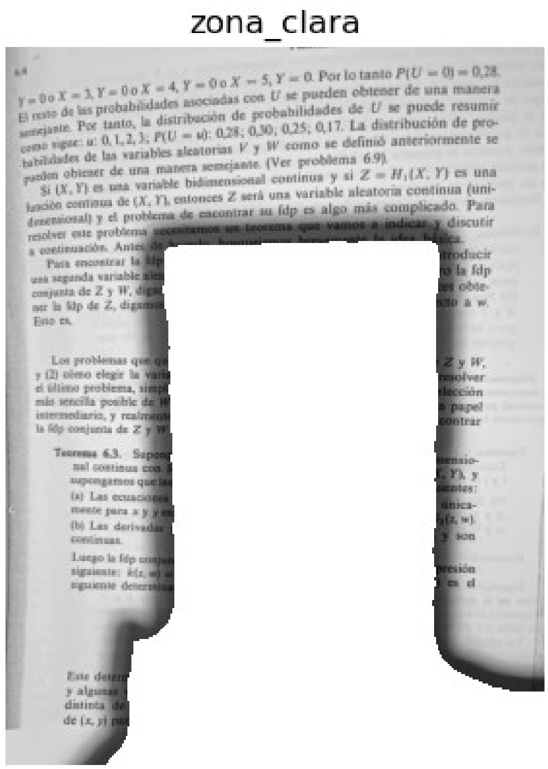
dan son (1) cómo encontrar la fdp conjunta d
ble aleatoria apropiada $W = H_2(X, Y)$. Para
mente, digamos que generalmente hacemos la
En el contexto presente, $W$ desempeña sólo u
no nos interesa en sí misma. Con el fin de e
necesitamos el teorema 6.3.

amos que $(X, Y)$ es una variable aleatoria bid
fdp conjunta $f$. Sea $Z = H_1(X, Y)$ y $W = H_2($
funciones $H_1$ y $H_2$ satisfacen las condiciones sig
$z = H_1(x, y)$ y $w = H_2(x, y)$ se pueden resolve
función de $z$ y $w$, digamos $x = G_1(z, w)$ y $y =$
parciales $\partial x/\partial z$, $\partial x/\partial w$, $\partial y/\partial z$, y $\partial y/\partial w$ existe

a de $(Z, W)$, digamos $k(z, w)$, está dada por la e
$f[G_1(z, w),\ G_2(z, w)]\, |J(z, w)|$, en donde $J(z,$
nte $2 \times 2$:

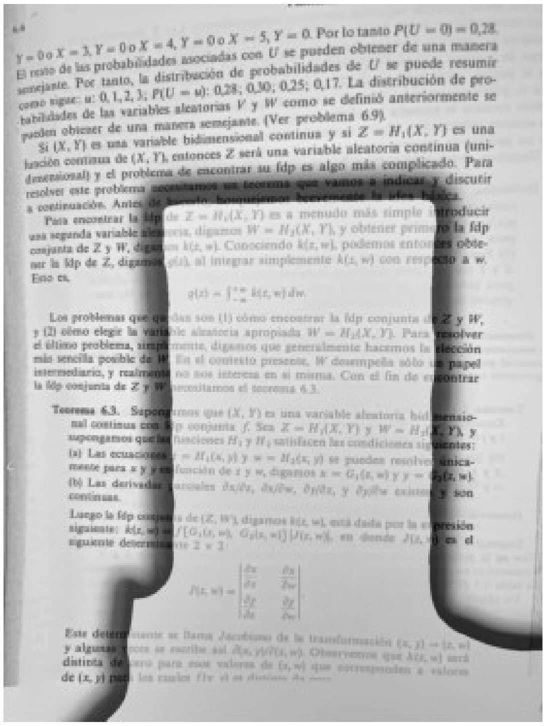
$$J(z, w) = \begin{vmatrix} \dfrac{\partial x}{\partial z} & \dfrac{\partial x}{\partial w} \\ \dfrac{\partial y}{\partial z} & \dfrac{\partial y}{\partial w} \end{vmatrix}.$$

inante se llama *Jacobiano* de la transformación $(x, y) \to (z, w)$
ces se escribe así $\partial(x, y)/\partial(z, w)$. Observemos que $k(z, w)$ será
cero para esos valores de $(z, w)$ que corresponden a valores
los cuales $f(x, y)$ es distinta de



```python
z1 =  np.nan_to_num(zona_clara,0)
z2 = np.nan_to_num(osc,0)

suma_parcial = (z1 + z2)
imprimir_pic(suma_parcial,"combianadas")
```

combianadas


6.4

$Y = 0$ o $X = 3, Y = 0$ o $X = 4, Y = 0$ o $X = 5, Y = 0$. Por lo tanto $P(U = 0) = 0{,}28$. El resto de las probabilidades asociadas con $U$ se pueden obtener de una manera semejante. Por tanto, la distribución de probabilidades de $U$ se puede resumir como sigue: u: 0, 1, 2, 3; $P(U = u)$: 0,28; 0,30; 0,25; 0,17. La distribución de probabilidades de las variables aleatorias $V$ y $W$ como se definió anteriormente se pueden obtener de una manera semejante. (Ver problema 6.9).

Si $(X, Y)$ es una variable bidimensional continua y si $Z = H_1(X, Y)$ es una función continua de $(X, Y)$, entonces $Z$ será una variable aleatoria continua (unidimensional) y el problema de encontrar su fdp es algo más complicado. Para resolver este problema necesitamos un teorema que vamos a indicar y discutir a continuación. Antes de hacerlo, bosquejemos brevemente la idea básica.

Para encontrar la fdp de $Z = H_1(X, Y)$ es a menudo más simple introducir una segunda variable aleatoria, digamos $W = H_2(X, Y)$, y obtener primero la fdp conjunta de $Z$ y $W$, digamos $k(z, w)$. Conociendo $k(z, w)$, podemos entonces obtener la fdp de $Z$, digamos $g(z)$, al integrar simplemente $k(z, w)$ con respecto a $w$. Esto es,

$$g(z) = \int_{-\infty}^{+\infty} k(z, w)\,dw.$$

Los problemas que quedan son (1) cómo encontrar la fdp conjunta de $Z$ y $W$, y (2) cómo elegir la variable aleatoria apropiada $W = H_2(X, Y)$. Para resolver el último problema, simplemente, digamos que generalmente hacemos la elección más sencilla posible de $W$. En el contexto presente, $W$ desempeña sólo un papel intermediario, y realmente no nos interesa en sí misma. Con el fin de encontrar la fdp conjunta de $Z$ y $W$ necesitamos el teorema 6.3.

**Teorema 6.3.** Supongamos que $(X, Y)$ es una variable aleatoria bidimensional continua con fdp conjunta $f$. Sea $Z = H_1(X, Y)$ y $W = H_2(X, Y)$, y supongamos que las funciones $H_1$ y $H_2$ satisfacen las condiciones siguientes:
(a) Las ecuaciones $z = H_1(x, y)$ y $w = H_2(x, y)$ se pueden resolver únicamente para $x$ y $y$ en función de $z$ y $w$, digamos $x = G_1(z, w)$ y $y = G_2(z, w)$.
(b) Las derivadas parciales $\partial x/\partial z$, $\partial x/\partial w$, $\partial y/\partial z$, y $\partial y/\partial w$ existen y son continuas.

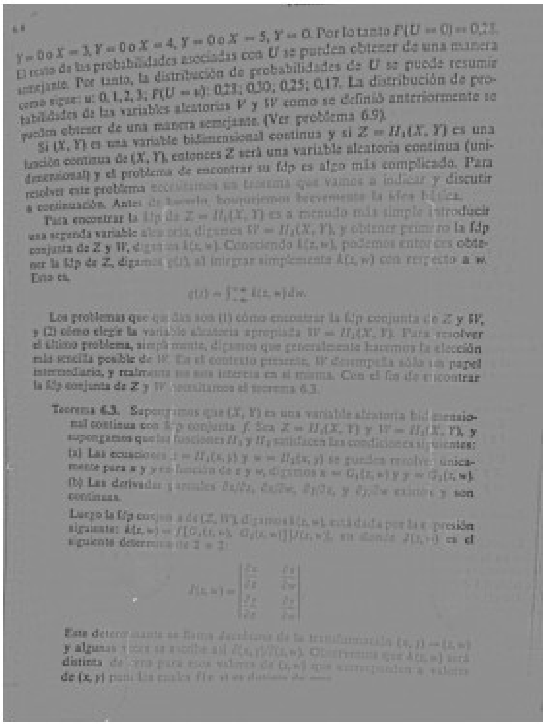
Luego la fdp conjunta de $(Z, W)$, digamos $k(z, w)$, está dada por la expresión siguiente: $k(z, w) = f[G_1(z, w), G_2(z, w)]\,|J(z, w)|$, en donde $J(z, w)$ es el siguiente determinante 2 × 2:

$$J(z, w) = \begin{vmatrix} \frac{\partial x}{\partial z} & \frac{\partial x}{\partial w} \\ \frac{\partial y}{\partial z} & \frac{\partial y}{\partial w} \end{vmatrix}$$

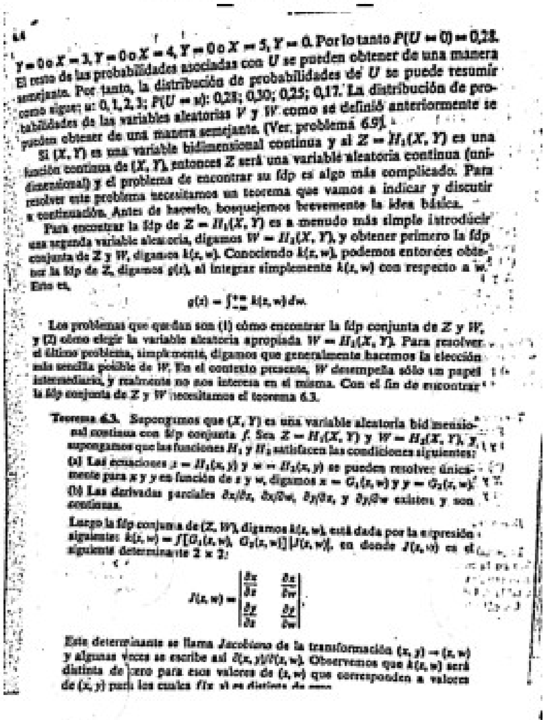
Este determinante se llama *Jacobiano* de la transformación $(x, y) \to (z, w)$ y algunas veces se escribe así $\partial(x, y)/\partial(z, w)$. Observemos que $k(z, w)$ será distinta de cero para esos valores de $(z, w)$ que corresponden a valores de $(x, y)$ para los cuales $f(x, y)$ es distinta de cero.

```
resta_parcial = np.copy(suma_parcial)

mask_gris = ndimage.median_filter(suma_parcial,
size=15,mode='reflect')


resta_parcial = ((suma_parcial - mask_gris)+255)/(2)
lessThen0 = resta_parcial<0
moreThen255 = resta_parcial>255
resta_parcial[lessThen0] = 0
resta_parcial[moreThen255] = 255


imprimir_pic(resta_parcial,"resta")
p_indicado = contraste_optimo_por_MSE(255 - resta_parcial,255 -
input_pic)
final_metodo_A = funcion_de_contraste(resta_parcial,p_indicado)
imprimir_pic(final_metodo_A,"final_metodo_A")
```

resta



final_metodo_A



6.4 $Y = 0$ o $X = 5$, $Y = 0$. Por lo tanto $P(U = 0) = 0{,}28$.
$Y = 0$ o $X = 3$, $Y = 0$ o $X = 4$, $Y = 0$ o
El resto de las probabilidades asociadas con $U$ se pueden obtener de una manera semejante. Por tanto, la distribución de probabilidades de $U$ se puede resumir como sigue: u: 0, 1, 2, 3; $P(U = u)$: 0,28; 0,30; 0,25; 0,17. La distribución de probabilidades de las variables aleatorias $V$ y $W$ como se definió anteriormente se pueden obtener de una manera semejante. (Ver problema 6.9).

Si $(X, Y)$ es una variable bidimensional continua y si $Z = H_1(X, Y)$ es una función continua de $(X, Y)$, entonces $Z$ será una variable aleatoria continua (unidimensional) y el problema de encontrar su fdp es algo más complicado. Para resolver este problema necesitamos un teorema que vamos a indicar y discutir a continuación. Antes de hacerlo, bosquejemos brevemente la idea básica.

Para encontrar la fdp de $Z = H_1(X, Y)$ es a menudo más simple introducir una segunda variable aleatoria, digamos $W = H_2(X, Y)$, y obtener primero la fdp conjunta de $Z$ y $W$, digamos $k(z, w)$. Conociendo $k(z, w)$, podemos entonces obtener la fdp de $Z$, digamos $g(z)$, al integrar simplemente $k(z, w)$ con respecto a $w$. Esto es,

$$g(z) = \int_{-\infty}^{+\infty} k(z, w)\,dw.$$

Los problemas que quedan son (1) cómo encontrar la fdp conjunta de $Z$ y $W$, y (2) cómo elegir la variable aleatoria apropiada $W = H_2(X, Y)$. Para resolver el último problema, simplemente, digamos que generalmente hacemos la elección más sencilla posible de $W$. En el contexto presente, $W$ desempeña sólo un papel intermediario, y realmente no nos interesa en sí misma. Con el fin de encontrar la fdp conjunta de $Z$ y $W$ necesitamos el teorema 6.3.

**Teorema 6.3.** Supongamos que $(X, Y)$ es una variable aleatoria bidimensional continua con fdp conjunta $f$. Sea $Z = H_1(X, Y)$ y $W = H_2(X, Y)$, y supongamos que las funciones $H_1$ y $H_2$ satisfacen las condiciones siguientes:
(a) Las ecuaciones $z = H_1(x, y)$ y $w = H_2(x, y)$ se pueden resolver únicamente para $x$ y $y$ en función de $z$ y $w$, digamos $x = G_1(z, w)$ y $y = G_2(z, w)$.
(b) Las derivadas parciales $\partial x/\partial z$, $\partial x/\partial w$, $\partial y/\partial z$, y $\partial y/\partial w$ existen y son continuas.

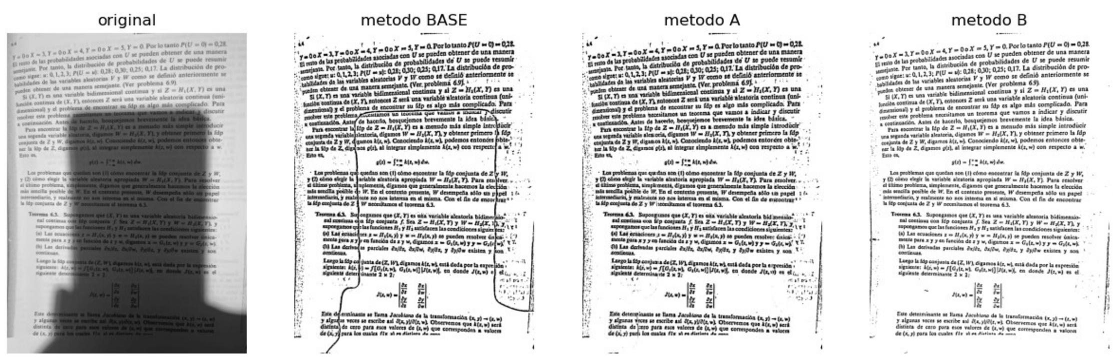
Luego la fdp conjunta de $(Z, W)$, digamos $k(z, w)$, está dada por la expresión siguiente: $k(z, w) = f[G_1(z, w), G_2(z, w)]\,|J(z, w)|$, en donde $J(z, w)$ es el siguiente determinante $2 \times 2$:

$$J(z, w) = \begin{vmatrix} \dfrac{\partial x}{\partial z} & \dfrac{\partial x}{\partial w} \\ \dfrac{\partial y}{\partial z} & \dfrac{\partial y}{\partial w} \end{vmatrix}.$$

Este determinante se llama *Jacobiano* de la transformación $(x, y) \to (z, w)$ y algunas veces se escribe así $\partial(x, y)/\partial(z, w)$. Observemos que $k(z, w)$ será distinta de cero para esos valores de $(z, w)$ que corresponden a valores de $(x, y)$ para los cuales $f(x, y)$ es distinta de cero.

```
plt.figure(figsize=(15,12))
plt.subplot(1,4, 1)
```

```python
plt.imshow(input_pic, cmap = 'gray')
plt.title('original')
plt.axis('off')

plt.subplot(1,4, 2)
plt.imshow(final_metodo_BASE, cmap = 'gray')
plt.axis('off')
plt.title('metodo BASE')



plt.subplot(1,4, 3)
plt.imshow(final_metodo_A, cmap = 'gray')
plt.axis('off')
plt.title('metodo A')

plt.subplot(1,4, 4)
plt.imshow(threshold_image, cmap = 'gray')
plt.axis('off')
plt.title('metodo B')
plt.show()
```




*Guardamos el mejor resultado como archivo jpeg*

```python
type(threshold_image)
```

```
PIL.Image.Image
```

```python
nombre = imagen_entrada + "_PROCESADA.jpeg"
```

```python
threshold_image.save(nombre)
```