



# TRABAJO PRÁCTICO N°1

## *“Procesamiento de Imágenes 1”*

### **Integrantes:**

- Cicoria, Ignacio
- Ricci, Guillermo

### **Profesores:**

- Sad, Gonzalo
- Julian Alvarez
- Juan Manuel Calle

### **Fecha entrega:**

- 21/04/2025

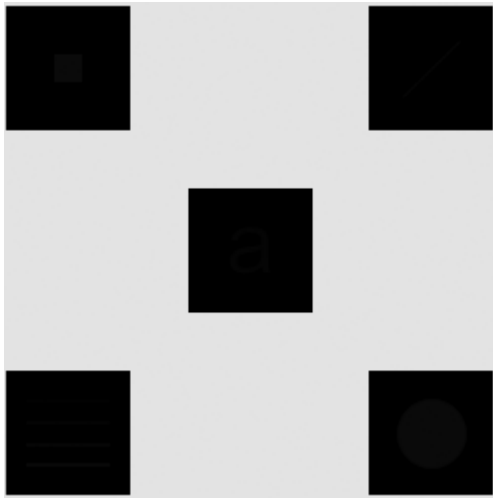
<b>Ecualización local de histograma.....</b>	<b>1</b>
1.1 Introducción.....	1
1.2 Ecualización Local.....	2
1.2.1 Impacto del Tamaño de la Ventana.....	3
1.3 Filtro de Mediana.....	4
1.3.2 Impacto del Tamaño del kernel.....	5
1.4 Conclusión.....	5
<b>Corrección de Exámenes Multiple Choice.....</b>	<b>6</b>
2.1 Introducción.....	6
2.2: Desarrollo del Algoritmo.....	7
2.2.1 Validación del Encabezado.....	7
2.2.2 Segmentación de Preguntas y Respuestas.....	7
2.2.3 Corrección Automática.....	8
2.2.4 Aplicación del Algoritmo a Conjunto de Exámenes.....	9
2.3. Generación de Imagen de Resultados Finales.....	10
2.4. Limitaciones y soluciones aplicadas.....	10
2.5. Conclusión.....	11
Información adicional.....	11
<b>Referencias.....</b>	<b>11</b>

# Ecualización local de histograma

## 1.1 Introducción

Para este trabajo, se utilizó una imagen en formato TIFF en escala de grises. La imagen contiene distintas figuras dentro de los cuadrados negros que por su tono son casi imperceptibles.

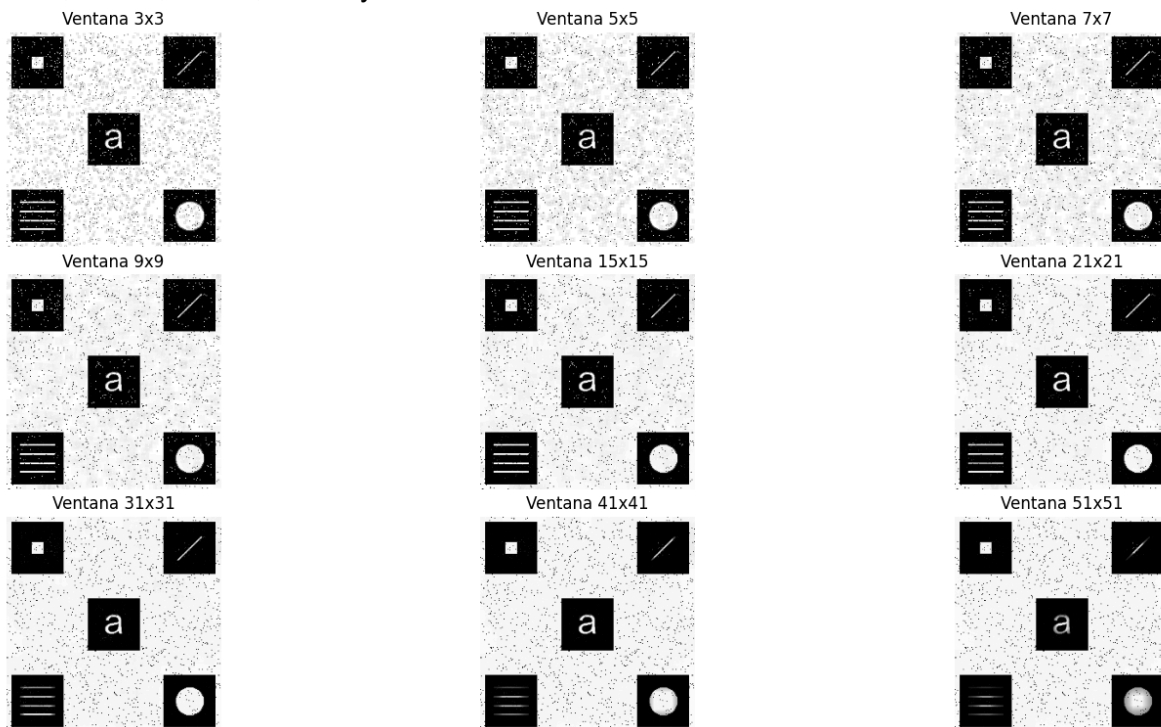
En el presente trabajo analizaremos el resultado de la ecualización con ventanas de diferentes tamaños e intentaremos un resultado de imagen que permita visualizar las figuras dentro de los cuadrados.



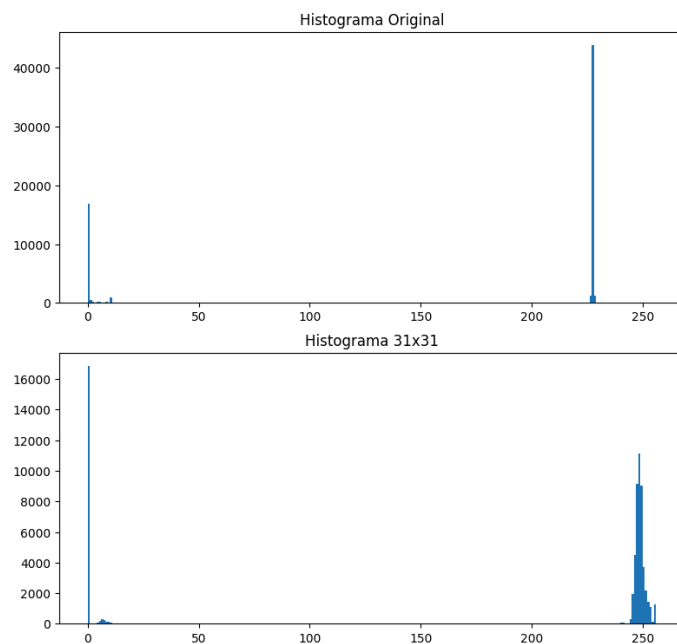
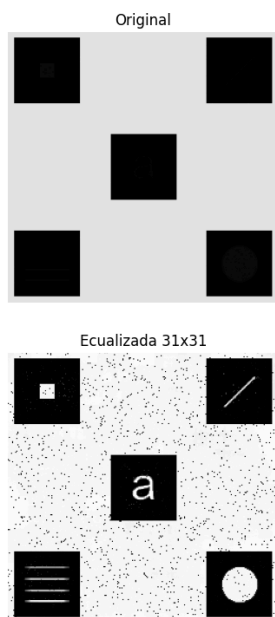
*imagen original*

## 1.2 Ecualización Local

Para la ecualización local se probaron diferentes tamaños de ventana: 3x3, 5x5, 7x7, 9x9, 15x15, 21x21, 31x31, 41x41 y 51x51.



*imagenes ecualizadas*



*Histograma de la Imagen Original y Ecualizada*

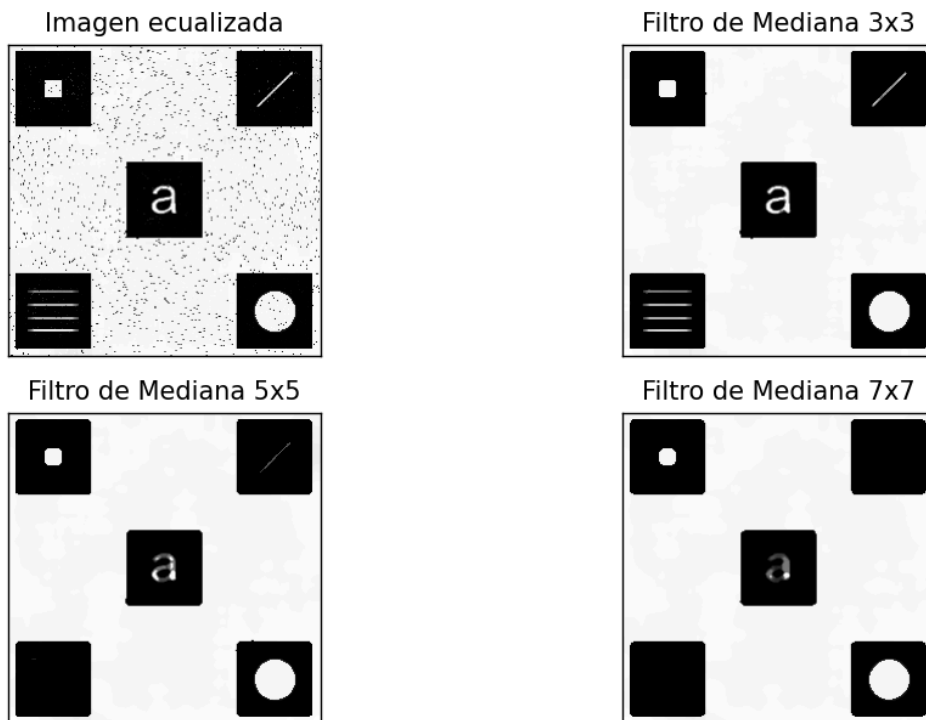
### 1.2.1 Impacto del Tamaño de la Ventana

El análisis de los diferentes tamaños de ventana ha demostrado que existe un equilibrio entre mejorar el contraste y preservar los detalles importantes en la imagen. A continuación se describe cómo afecta cada tamaño de ventana:

- **Ventanas pequeñas (3x3, 5x5, 7x7, 9x9, 15x15, 21x21):** Estas ventanas tienden a introducir ruido. Aunque mejoran el contraste local, afectan la coherencia de la imagen.
- **Ventanas medianas (31x31 y 41x41):** Las ventanas de 31x31 y 41x41 muestran un buen equilibrio, mejorando el contraste sin perder detalles importantes. Estas dimensiones permiten visualizar correctamente las figuras que se quieren resaltar y a su vez reducen el ruido de la imagen.
- **Ventanas grandes (51x51):** La ventana tiende a suavizar excesivamente las figuras a resaltar, haciendo que se pierdan detalles importantes y se difuminen bordes.

## 1.3 Filtro de Mediana

A su vez aplicamos filtros de mediana sobre la imagen ecualizada con diferentes tamaños de kernel: 3x3, 5x5 y 7x7.



*imágenes con filtros de mediana*

### 1.3.2 Impacto del Tamaño del kernel

Podemos observar como el kernel de menor tamaño, 3x3, genera un resultado más equilibrado reduciendo el ruido de la imagen casi en su totalidad y preservando las figuras de interés. Los de mayor tamaño reducen más el ruido pero suavizan las figuras y eliminan algunas de ellas.

## 1.4 Conclusión

A partir del análisis realizado, se concluye que la ecualización local del histograma permite mejorar significativamente el contraste en imágenes, facilitando la visualización de detalles que inicialmente eran imperceptibles. Sin embargo, el tamaño de la ventana utilizada es determinante en el resultado: ventanas pequeñas tienden a introducir ruido, mientras que ventanas muy grandes suavizan excesivamente la imagen y eliminan información relevante.

Las dimensiones de 31x31 y 41x41 ofrecen un equilibrio óptimo entre mejora del contraste y preservación de detalles importantes.

Asimismo, la aplicación del filtro de mediana sobre las imágenes ecualizadas demostró ser efectiva para reducir el ruido sin comprometer excesivamente la calidad de la imagen. El kernel de 3x3 resultó ser el más adecuado, ya que elimina ruido conservando las figuras clave. Los kernels más grandes, si bien reducen más el ruido, también tienden a borrar o suavizar demasiado las estructuras relevantes.

En conjunto, el uso combinado de una ecualización local con ventana de tamaño 31x31 y un filtro de mediana de 3x3 resultó una estrategia eficaz para realzar detalles ocultos y mejorar la percepción de la imagen.

## Corrección de Exámenes Multiple Choice

### 2.1 Introducción

El objetivo del presente trabajo práctico es desarrollar un algoritmo capaz de corregir automáticamente exámenes tipo múltiple choice a partir de imágenes escaneadas. Cada examen cuenta con 25 preguntas de opción múltiple y un encabezado con datos personales. Se busca evaluar tanto las respuestas como la validez del encabezado, y generar un informe con los resultados y desempeño de cada alumno.

Se proporcionan imágenes de exámenes resueltos y una plantilla de respuestas correctas. Las respuestas válidas son aquellas en las que se ha seleccionado una única opción por pregunta. Las preguntas con más de una opción marcada o sin ninguna opción son consideradas incorrectas.

Además, el encabezado debe ser validado conforme a los siguientes criterios:

- **Name:** Debe contener al menos dos palabras y no más de 25 caracteres.
- **ID:** Exactamente 8 caracteres, sin espacios.
- **Code:** Un único carácter.
- **Date:** 8 caracteres en total.

## 2.2: Desarrollo del Algoritmo

### 2.2.1 Validación del Encabezado

Se segmentan los campos Name, ID, Code y Date, y se verifica que cumplan con las restricciones.

*Ejemplo de encabezado:*

Name	ID	Code	Date
Pedro Monti		A	23-033-24

*Ejemplo de salida esperada:*

```
Evaluando examen 3
Resultados de validación:
Name: OK
ID: MAL
Code: OK
Date: MAL
```

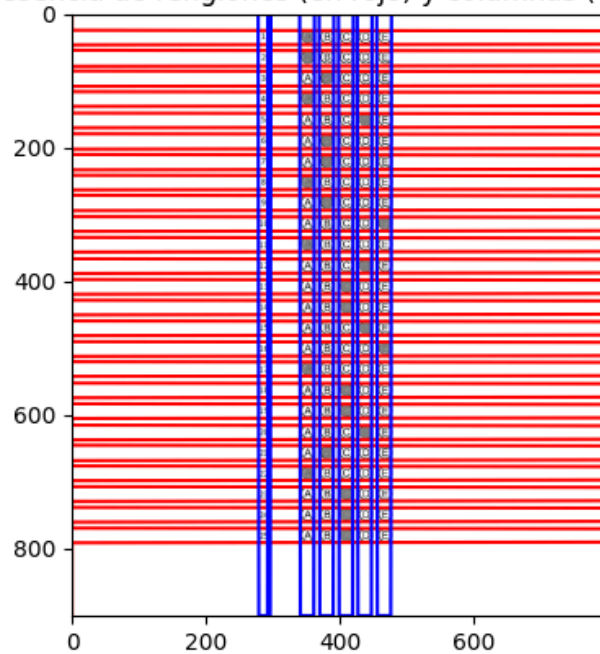
### 2.2.2 Segmentación de Preguntas y Respuestas

Para detectar automáticamente las preguntas y respuestas marcadas, primero se recorta la región donde se encuentran las respuestas. A esa zona se le aplica un umbral de binarización, considerando como marcas válidas los píxeles con valor menor a 240. Luego, se analizan todas las filas y columnas de la imagen para detectar cuáles contienen al menos un píxel negro. Esto permite determinar la ubicación de los renglones que representan las preguntas y de las columnas correspondientes a las opciones de respuesta.

A partir de estos datos se calculan los límites de cada renglón y de cada columna. En cada renglón se ignoran las dos primeras columnas, y se recortan las imágenes correspondientes a las opciones de respuesta. Para determinar si una casilla fue marcada, se calcula el porcentaje de píxeles negros sobre el total. Si ese valor supera un umbral predeterminado (en este caso 0.8), se interpreta que esa opción fue seleccionada por el estudiante. Finalmente, se asigna la letra correspondiente (A, B, C, D o E) a la columna marcada y se almacena la información de las respuestas en un diccionario.



Presencia de renglones (en rojo) y columnas (en azul)



### 2.2.3 Corrección Automática

Se comparan las respuestas detectadas con la plantilla de respuestas correctas:

```

respuestas_correctas = ['A', 'A', 'B', 'A', 'D', 'B', 'B', 'C', 'B', 'A',
                        'D', 'A', 'C', 'C', 'D', 'B', 'A', 'C', 'C', 'D',
                        'B', 'A', 'C', 'C', 'C']
  
```

Corrección del examen:

Pregunta 1: ✓ Correcta (A)

Pregunta 2: ✓ Correcta (A)

Pregunta 3: ✓ Correcta (B)

Pregunta 4: ✗ Incorrecta (E → correcta: A)

Pregunta 5: ✗ Incorrecta (A → correcta: D)

Pregunta 6: ✗ Incorrecta (A → correcta: B)

Pregunta 7: ✗ Incorrecta (D → correcta: B)

Pregunta 8: ✗ Incorrecta (A → correcta: C)

Pregunta 9: ✗ Incorrecta (A → correcta: B)

Pregunta 10: ✗ Incorrecta (E → correcta: A)

Pregunta 11: ✓ Correcta (D)

Pregunta 12: ✗ Incorrecta (D → correcta: A)

Pregunta 13: ✓ Correcta (C)

Pregunta 14: ✗ Incorrecta (D → correcta: C)

Pregunta 15: ✗ Incorrecta (B → correcta: D)

Pregunta 16: ✗ Incorrecta (E → correcta: B)

Pregunta 17: ✗ Incorrecta (D → correcta: A)

Pregunta 18: ✗ Sin respuesta o ambigua

Pregunta 19: ✗ Incorrecta (B → correcta: C)

Pregunta 20: ✗ Incorrecta (C → correcta: D)

Pregunta 21: ✗ Incorrecta (D → correcta: B)

Pregunta 22: ✗ Incorrecta (C → correcta: A)

Pregunta 23: ✗ Incorrecta (D → correcta: C)

Pregunta 24: ✗ Incorrecta (E → correcta: C)

Pregunta 25: ✗ Incorrecta (E → correcta: C)

Total de aciertos: 5/25

✗ El alumno está DESAPROBADO

*Salida en pantalla de cada examen*

## 2.2.4 Aplicación del Algoritmo a Conjunto de Exámenes

Se procesan cinco imágenes de exámenes, aplicando el algoritmo completo. Se reporta para cada alumno:

- Total de respuestas correctas
- Estado de cada campo del encabezado
- Condición: Aprobado ( $\geq 20$  respuestas correctas) o Desaprobado

## 2.3. Generación de Imagen de Resultados Finales

Se crea una imagen final que incluye:

- Recorte del campo **Name** de cada examen.
- Un indicador visual (por ejemplo, color verde para aprobado, rojo para desaprobado).

Alumno 1 APROBADO	Alumno 2 DESAPROBADO	Alumno 3 DESAPROBADO	Alumno 4 DESAPROBADO	Alumno 5 DESAPROBADO
Juan Perez	Jorge	Pedro Monti	Alfredo	Maria Suarez

## 2.4. Limitaciones y soluciones aplicadas

En el desarrollo del algoritmo nos fuimos encontrando ante diferentes problemas y tomando distintas decisiones para solucionarlos.

El área para determinar los componentes válidos se seteó en 3 para que lo menor que detecte sean los guiones.

```
def contar_componentes_validas(img_bin, th_area=3)
```

Tanto el encabezado como la imagen de las respuestas se recortaron de forma manual. Es posible tomar esta decisión porque el encabezado en los exámenes a evaluar se encuentra en la misma posición. Para la imagen de respuestas únicamente cortamos el encabezado. Para el análisis de otros exámenes debería evaluarse que el encabezado esté dentro de las coordenadas establecidas y de no ser así determinar las coordenadas que permitan realizar el mismo recorte en todos los exámenes.

```
encabezado = img[102:135, 22:775]
```

```
img_respuestas = img[155:1056, 10:810]
```

Los umbrales de detección de líneas se definieron visualizando la longitud de las líneas del encabezado, diferenciando las verticales de las horizontales.

```
umbral_vertical = 25
umbral_horizontal = 300
```

Debido a que la imagen de los campos del encabezado contiene la primer línea vertical y horizontal del mismo decidimos correr el inicio del mismo

```
# Coordenadas para recorte vertical (encabezado de una sola fila)
y_ini = lineas_horizontales[0]+3
```

```
x_ini = lineas_verticales[i]+2
```

Para que evalúe por renglón las columnas que representan una respuesta tomamos a partir de la tercer columna

```
columnas_idxes_reducidas = columnas_idxes[2:] # Saltamos las dos
primeras columnas
```

---

## 2.5. Conclusión

El desarrollo del algoritmo para la corrección automática de exámenes permitió automatizar con éxito el proceso de validación del encabezado, segmentación de respuestas, corrección y generación de una imagen final con los resultados.

Los resultados obtenidos son satisfactorios para el conjunto de exámenes evaluados. Sin embargo, el algoritmo presenta limitaciones en cuanto a su generalización: ante exámenes con variaciones en el diseño o posición de los campos, sería necesario adaptar dinámicamente el análisis de encabezado y respuestas.

## Referencias

1. <https://docs.python.org/3/library/venv.html>
2. <https://docs.python.org/es/3.13/reference/index.html>

