

Instituto Tecnológico de Costa Rica

CE1102 - Taller de introducción a la programación

Tarea Programada 1: Filtro de medianas

Profesor: Saúl Calderón Ramírez

Estudiantes: Manuel Calero Ríos, Brandon Mora Díaz y Fabricio Picado  
Alvarado

Primer semestre 2022

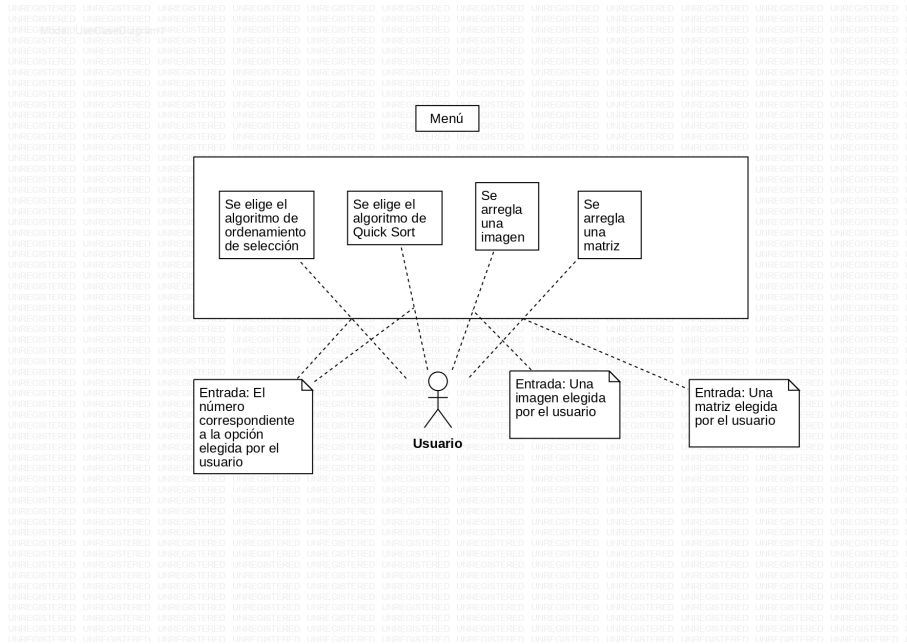
## Contents

<b>1</b>	<b>Introducción</b>	<b>3</b>
<b>2</b>	<b>Análisis del problema.</b>	<b>3</b>
<b>3</b>	<b>Diseño de la solución</b>	<b>4</b>
3.1	Pseudo-código . . . . .	4
<b>4</b>	<b>Implementación y pruebas.</b>	<b>6</b>
4.1	Prueba de filtrado de una matriz . . . . .	6
4.2	Prueba de filtrado de una matriz . . . . .	7
4.3	Prueba de filtrado de una matriz . . . . .	7
4.4	Prueba de filtrado de una imagen . . . . .	7
<b>5</b>	<b>Conclusiones y recomendaciones</b>	<b>9</b>
<b>6</b>	<b>Bibliografía</b>	<b>9</b>

# 1 Introducción

En ocasiones cuando se saca una fotografía a alguna persona u objeto, la imagen puede venir perjudicada por el ruido que ocasiona la transferencia de bytes que se pueden confundir en el camino. En esta ocasión se realizará un programa que puede solucionar los errores que ocasiona el ruido en la imágenes. Para comenzar, se utilizará una imagen en escala de grises a la cual se le realizarán sustituciones a los números que se consideran errores, cuyos valores serán 0 o 255. Para la sustitución de estos números se calculará la media y esta reemplazará estos números. La media es un valor intermedio que es determinado por los valores de la matriz. La matriz es un área, ya sea cuadrangular o rectangular, que es determinada por el usuario, mediante el uso de una ventana. Una vez realizado todo el proceso de intercambiado de números, se procede a realizar el guardado de la nueva imagen o matriz. Para que todo esto sea posible, se utilizarán tanto recursividad de pila y cola, la librería numpy y se hará uso del código Gray con el fin de que facilite el calculo de imágenes en escala de grises.

# 2 Análisis del problema.



Las entradas serán o bien, una imagen en blanco y negro o una matriz.

Subproblemas:

1. Preguntarle al usuario si lo que quiere transformar es una imagen o una matriz, en caso de seleccionar una matriz se pasaría directamente al subproblema 4
2. Convertir la imagen en escala de grises mediante el uso del código Gray
3. Con los datos obtenidos en escala de grises, se calcularán las matrices de las listas de listas que se ahí salen.
4. Preguntarle al usuario el tamaño de la ventana
5. La ventana deberá recolectar los datos que estén dentro de su límite
6. Los datos que fueron obtenidos, serán ordenados de manera ascendente
7. Se calculará una media con un valor intermedio entre los datos obtenidos anteriormente
8. Una vez ordenados los datos, se realizará el intercambio de los valores equivalentes a 0 o 255 por la media obtenida previamente
9. Realizar la transformación de los nuevos dígitos obtenidos a una imagen o una matriz (en caso de ser matriz, simplemente se le devuelve la nueva matriz al usuario)
10. Se le informa al usuario la duración del proceso
11. Se debe guardar el resultado

Restricciones:

1. La imagen no debe ser a color, sino en blanco y negro
2. Se debe calcular la media utilizando el valor del medio y no mediante la fórmula de media aritmética
3. Se deben utilizar los algoritmos de selección y de quick-sort

La salida será la imagen ya corregida o la matriz con sus debidas correcciones

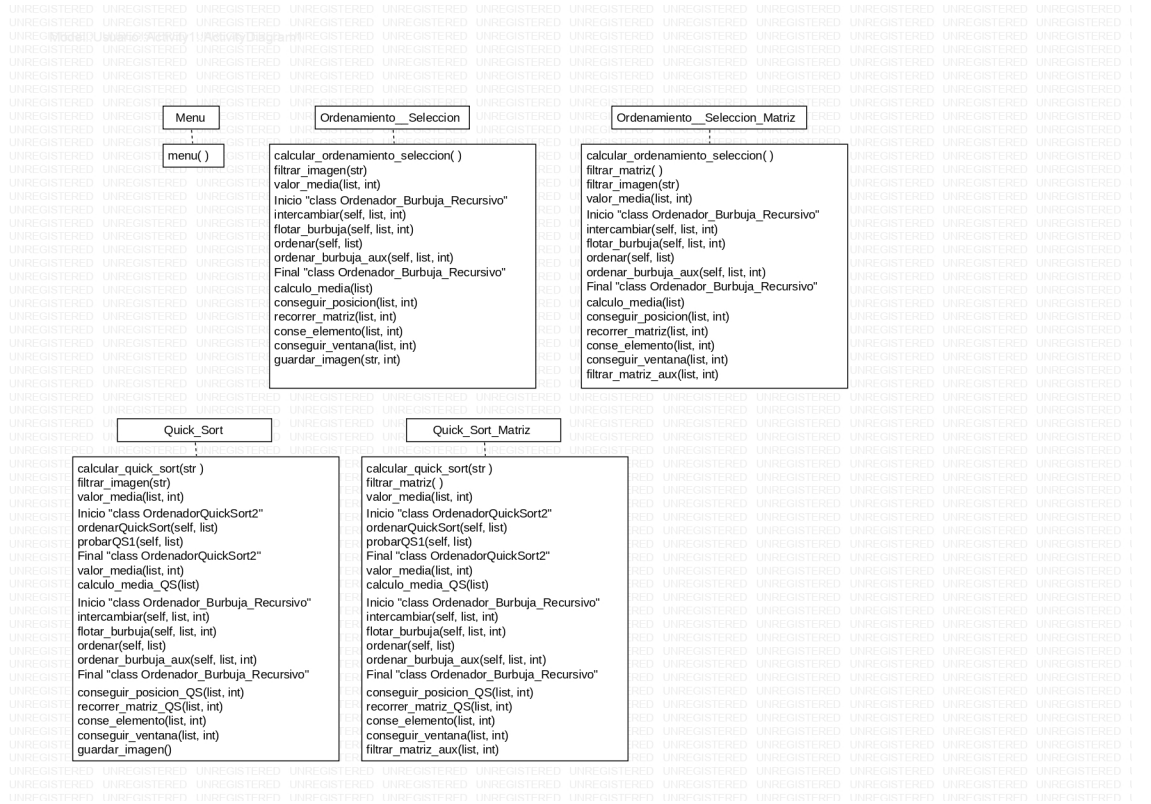
## 3 Diseño de la solución

### 3.1 Pseudo-código

1. Se le pregunta al usuario que tipo de algoritmo desea implementar
  - 1.1. Si el usuario selecciona el de ordenamiento por selección, ir al paso 2
  - 1.2. Si el usuario selecciona el de quick sort, ir al paso 3

2. El usuario seleccionar el algoritmo de ordenamiento por selección para calcular una imagen o una matriz
  - 2.1. Si selecciona una imagen, ir al paso 4
  - 2.2. Si selecciona una matriz, ir al paso 8
3. El usuario seleccionar el algoritmo de ordenamiento por selección para calcular una imagen o una matriz
  - 3.1. Si selecciona una imagen, ir al paso 4
  - 3.2. Si selecciona una matriz, ir al paso 8
4. Seleccionar la imagen
  - 4.1. Descomponer la imagen en código Gray en escala de grises
    - 4.1.1. Descomponer la imagen en listas mediante recursividad de listas
    - 4.1.2. Determinar los márgenes de las listas
    - 4.1.3. Determinar el tamaño de las ventanas
      - 4.1.3.1. Se ordenan, mediante el algoritmo que el usuario haya seleccionado, los elementos de las ventanas de manera ascendente y pasar al paso 3
5. Determinar la media
  - 5.1. Seleccionar el valor central de la ventana
6. Reemplazar los valores
  - 6.1. Si dentro de los valores ordenados en las ventanas existe uno que sea 0 o 255, este valor debe ser reemplazado por la media misma
7. Guardado
  - 5.1. Una vez procesada la nueva lista, esta se debe procesar como una imagen
  - 5.2. El usuario debe seleccionar el directorio donde desea guardar la imagen
  - 5.3. Se debe guardar la nueva imagen e informar al usuario el tiempo que tardó el proceso
8. El usuario ha seleccionado una matriz
  - 8.1. Se verifica la entrada
  - 8.2. Se le pregunta al usuario el tamaño de la ventana de la matriz

- 8.3. Se realiza el acomodamiento de las imágenes, usando el algoritmo que haya seleccionado el usuario
- 8.4. Se calcula la media de los números acomodados
- 8.5. Se le devuelve la nueva imagen al usuario junto con el tiempo que tardó el proceso



## 4 Implementación y pruebas.

### 4.1 Prueba de filtrado de una matriz

1. Se le da como entrada a la funcion *filtrar\_imagen(matriz, K, algoritmo\_ordenamiento)*, la siguiente *matriz*

```

matriz_3 = np.array([[221.0, 222, 250, 251, 223, 249],
[223.0, 220, 250, 251, 225, 242],
[221.0, 0, 250, 251, 221, 249],
[221.0, 222, 255, 251, 0, 249],
[220.0, 219, 250, 251, 221, 249],
[221.0, 222, 250, 251, 223, 249]])
  
```

Con un  $K = 3$ . Usando el algoritmo de ordenamiento seleccion.

## 4.2 Prueba de filtrado de una matriz

1. Se le da como entrada a la funcion *filtrar\_imagen(matriz, K, algoritmo\_ordenamiento)*, la siguiente *matriz*

```
matriz_3 = np.array([[221.0, 222, 250, 251, 223, 249],  
[223.0, 220, 250, 251, 225, 242],  
[221.0, 0, 250, 251, 221, 249],  
[221.0, 222, 255, 251, 0, 249],  
[220.0, 219, 250, 251, 221, 249],  
[221.0, 222, 250, 251, 223, 249]])
```

Con un  $K = 4$ . Usando el algoritmo de ordenamiento quick sort

## 4.3 Prueba de filtrado de una matriz

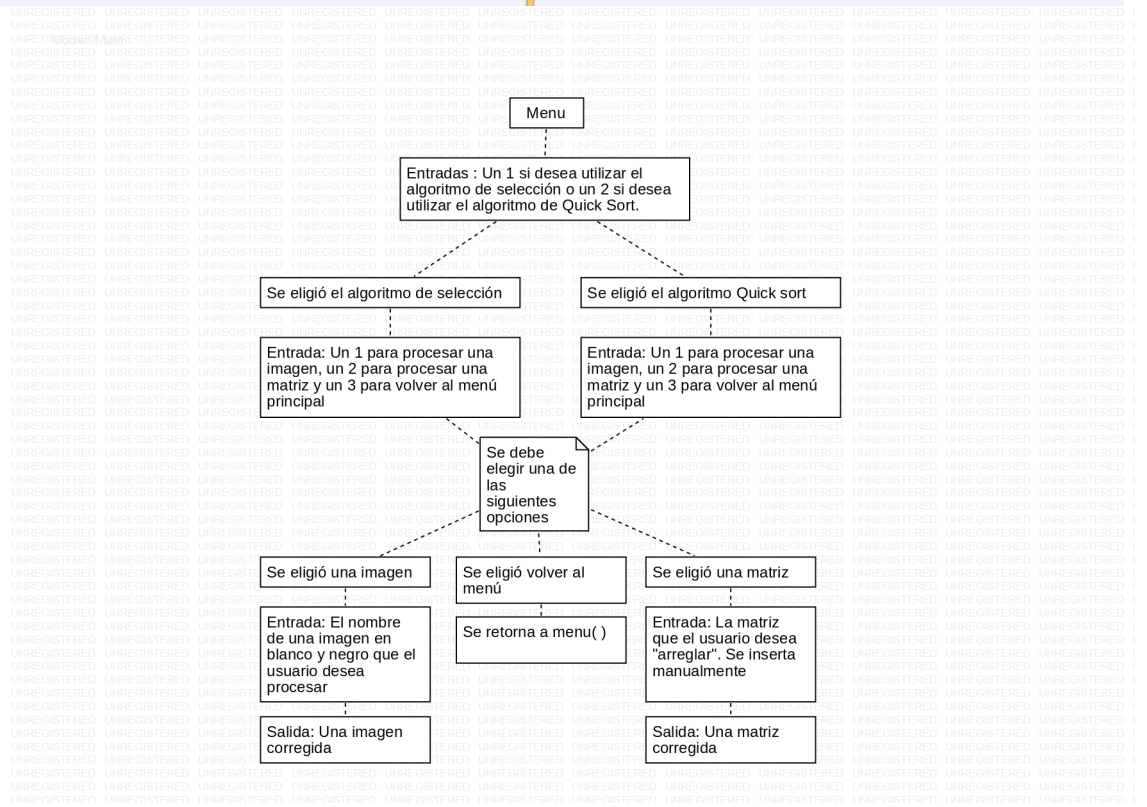
1. Se le da como entrada a la funcion *filtrar\_imagen(matriz, K, algoritmo\_ordenamiento)*, la siguiente *matriz*

```
matriz_3 = np.array([[221.0, 222, 250, 251, 223, 249],  
[223.0, 220, 250, 251, 225, 242],  
[221.0, 0, 250, 251, 221, 249],  
[221.0, 222, 255, 251, 0, 249],  
[220.0, 219, 250, 251, 221, 249],  
[221.0, 222, 250, 251, 223, 249]])
```

Con un  $K = 3$ . Usando el algoritmo de ordenamiento quick sort.

## 4.4 Prueba de filtrado de una imagen

1. Se da como entrada una imagen a la cual hay que eliminar los “ruidos” producidos en ella





## 5 Conclusiones y recomendaciones

1. Hubo instrucciones que se facilitaron por los temas vistos en las semanas anteriores como manipulación de listas para ordenarlas de manera ascendente y sacar la mediana.
2. Un reto fue el entendimiento completo de las matrices ya que se tuvo que recorrer una imagen, y en los fragmentos con ruido convertirlos a listas para solucionarlo.
3. El aspecto mas desafiante fue el de lograr que funcionará con diferentes tamaños de ventana, ya que conforme se hace mas grande se elige, mas pesado es el trabajo que necesita realizar.
4. Por otro lado mejoramos el trabajo en equipo ya que entre varios integrantes al realizar un trabajo más complejo se tuvo que designar roles y coordinar tiempos para hablar y mejorar el progreso que se llevaba.

## 6 Bibliografía

### References

- [1] Yolanda Martinez Treviño. Ejemplo Python: Crear una matriz como lista de listas. (1 de octubre de 2020). Accedido el 11 de mayo de 2022. [Video en línea]. Disponible: <https://www.youtube.com/watch?v=uJy3vPXzjs0>
- [2] DayiTecnologia. Matrices en Python --EJERCICIO Programacion RESUELTO--Matriz con listas anidadas. (13 de julio de 2021). Accedido el 11 de mayo de 2022. [Video en línea]. Disponible: [https://www.youtube.com/watch?v=\\_ohpNH2YQmY](https://www.youtube.com/watch?v=_ohpNH2YQmY)
- [3] Mimi. "Crear una imagen BGR, escala de gris openCV python". kipunaEc. <https://noemioocc.github.io/posts/Crear-una-imagen-BGR-openCV-Python/> (accedido el 11 de mayo de 2022).
- [4] R. Wachenchauser, M. Manterola, M. Curia, M. Medrano y N. Paez. "Ordenamiento por selección (Algoritmos de Programación con Python)". Diseño y programación web (libros, tutoriales y vídeos sobre HTML, CSS, JavaScript, PHP). <https://uniwebsidad.com/libros/algoritmos-python/capitulo-19/ordenamiento-por-seleccion> (accedido el 11 de mayo de 2022).
- [5] "QuickSort (With Code in Python/C++/Java/C)". Programiz: Learn to Code for Free. <https://www.programiz.com/dsa/quick-sort> (accedido el 11 de mayo de 2022).