Discusión de complejidad

Clase de Práctico N°6
Procesamiento Digital de Señales

INGENIERÍA
BIOLÓGICA

Contenidos

01

Revisión del Hito 2.2

Medición de complejidad sobre el FIR implementado

02

Repaso

Repaso de los contenidos vistos en el práctico: validación del filtro y Valgrind.

03

Entrega 2

Obtención y análisis de los resultados

01

Revisión del Hito 2.2

Medición de complejidad del FIR implementado

- Determinar de forma teórica la complejidad de su filtro.
- Utilizar callgrind para determinar de forma experimental la cantidad de ciclos de procesador que utiliza su filtro, para distintos valores de N y L.

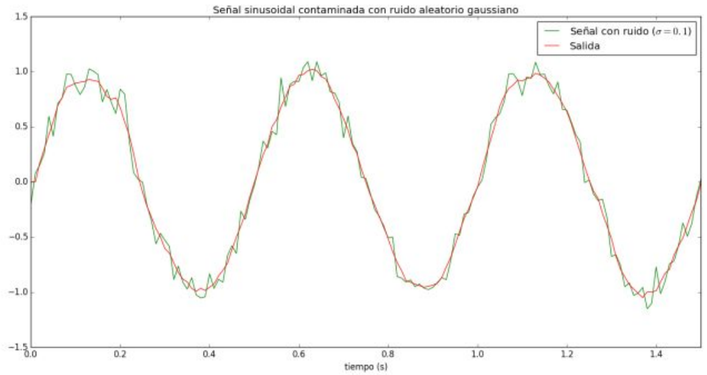
02

Repaso

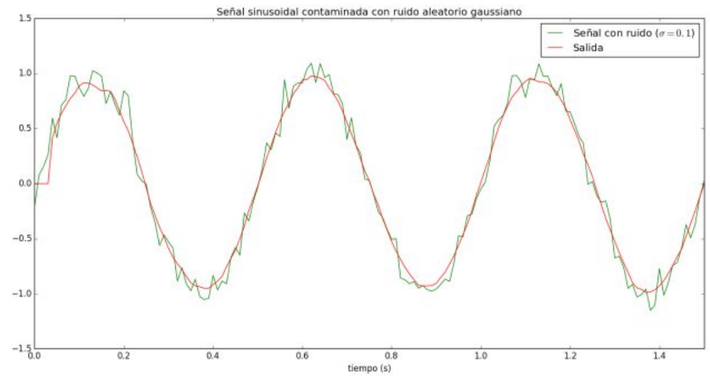
Repaso de los contenidos vistos en el práctico: validación del filtro y Valgrind.

Validación del filtro

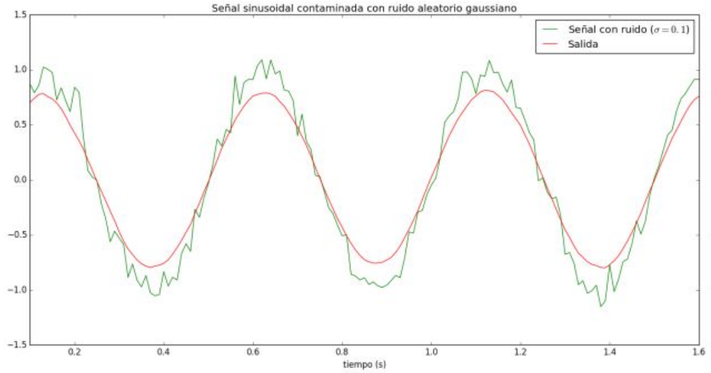
Análisis cualitativo



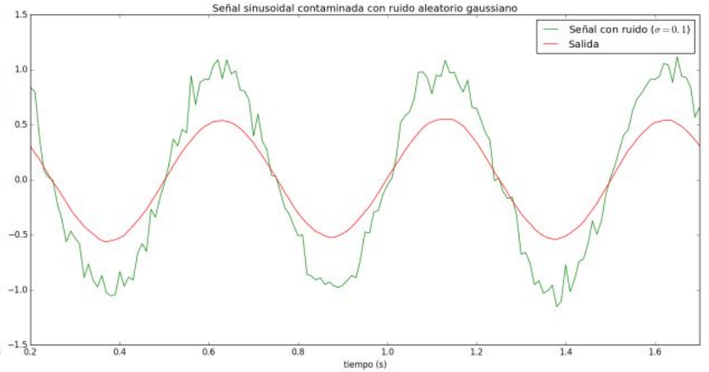
(a) Tamaño de filtro L=5



(b) Tamaño de filtro L=9



(c) Tamaño de filtro L=19



(d) Tamaño de filtro L=29

Validación del filtro

Respuesta al escalón

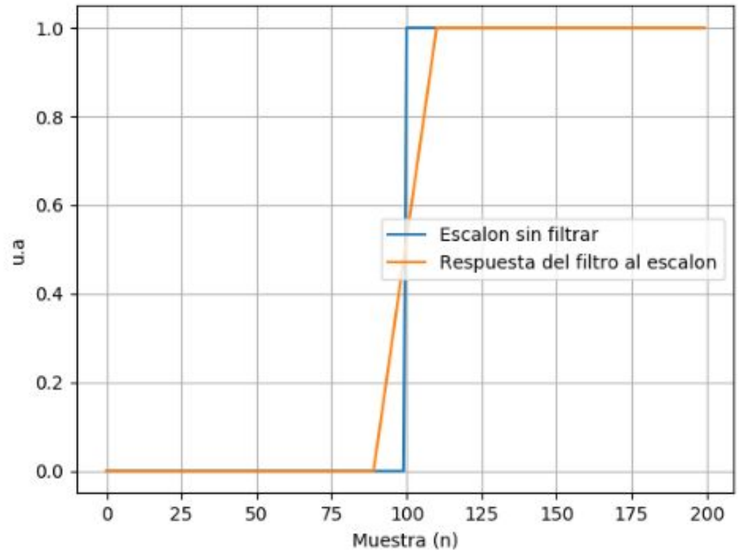
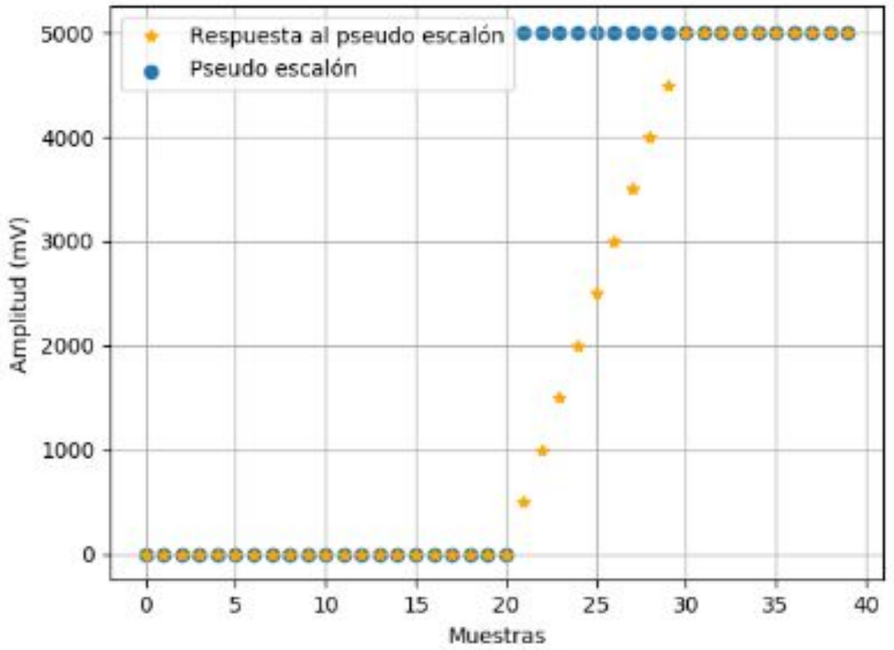


Figura 2: Respuesta del filtro al escalón.

FIR no causal



FIR causal

Validación del filtro

Respuesta en frecuencia

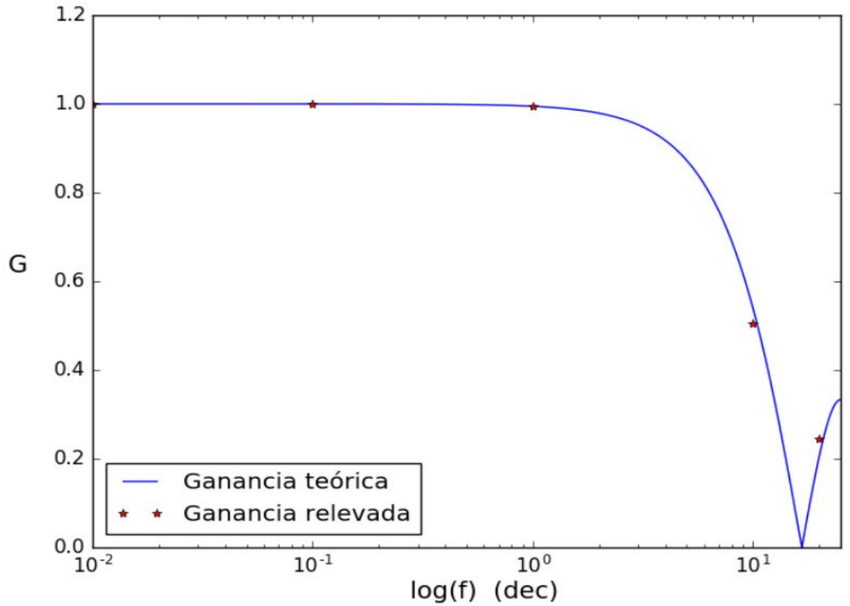
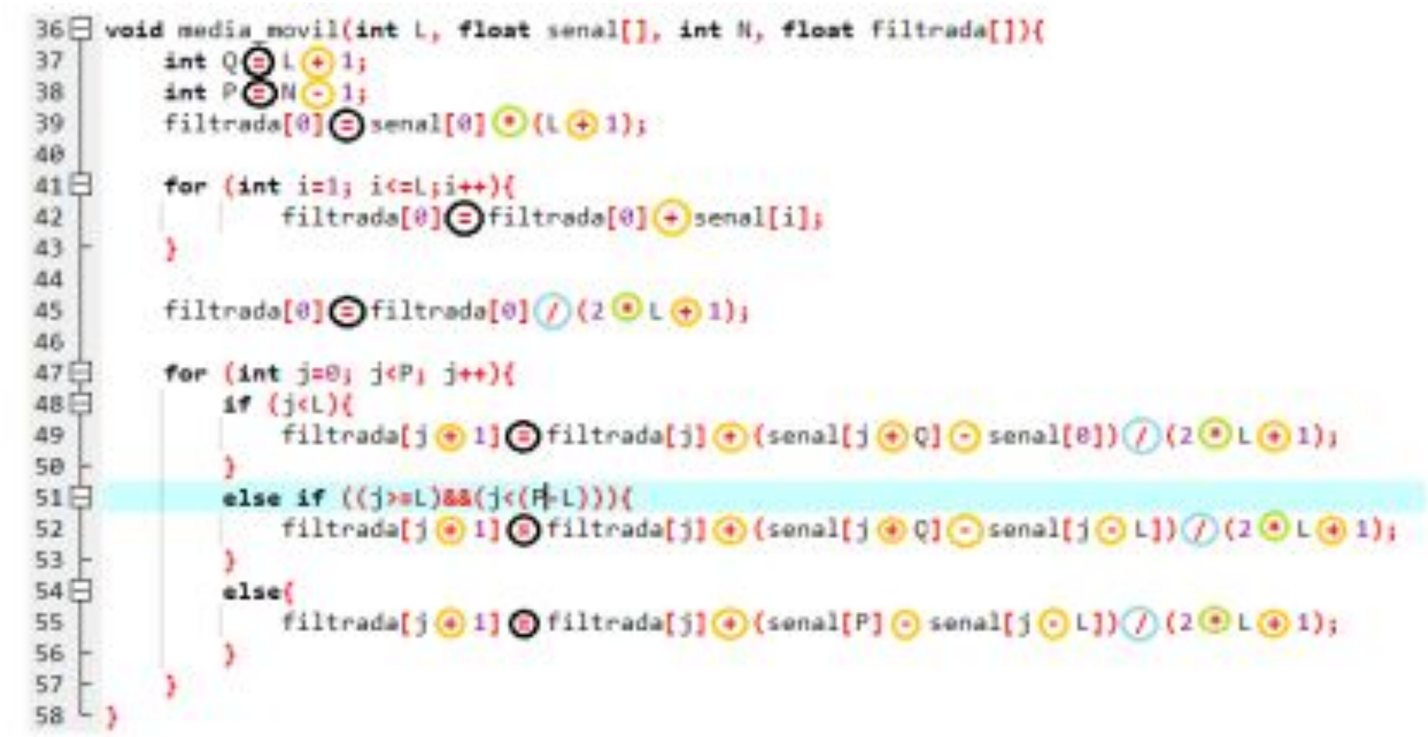


Figura 2: Módulo de la transferencia de la Media Móvil de largo 3. Los valores de frecuencia se expresan en décadas. Se relevó el comportamiento del filtro para valores de frecuencia $F \in [10mHz, 100mHz, 1Hz, 10Hz, 20Hz]$. En rojo se grafica la tupla (log(frecuencia), ganancia) de valores relevados en la práctica.

Cálculo de complejidad teórica



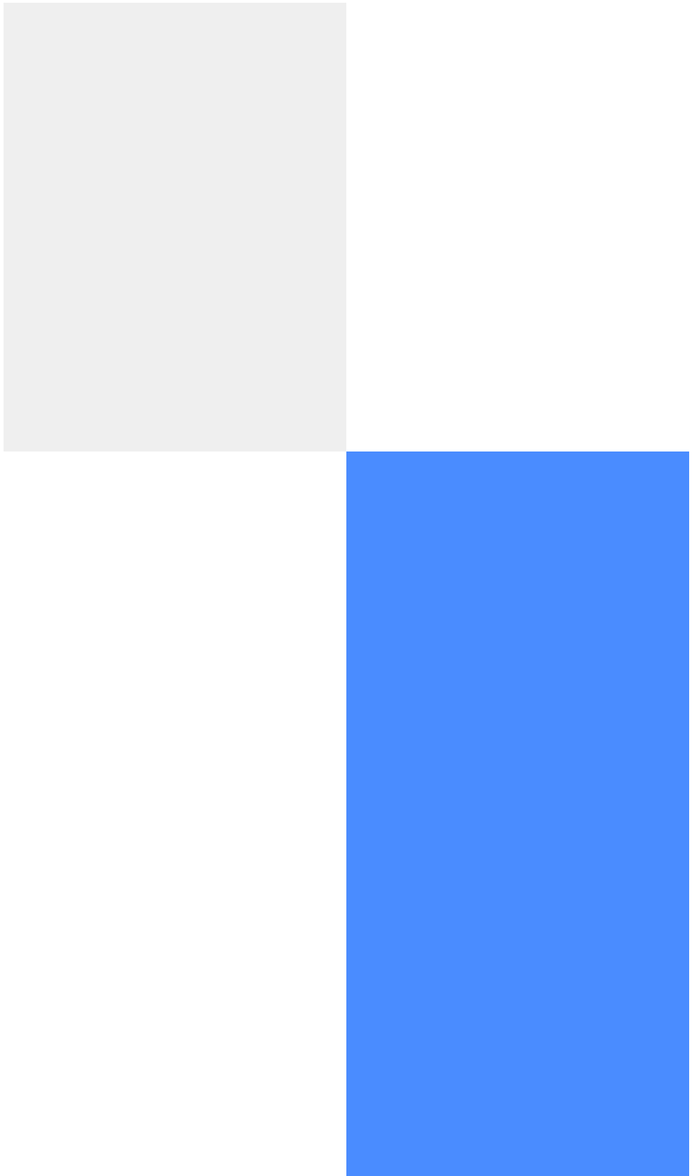
```
36  void media_movil(int L, float senal[], int N, float filtrada[]){
37      int Q = L + 1;
38      int P = N - 1;
39      filtrada[0] = senal[0] * (L + 1);
40
41      for (int i=1; i<=L;i++){
42          filtrada[0] = filtrada[0] + senal[i];
43      }
44
45      filtrada[0] = filtrada[0] / (2 * L + 1);
46
47      for (int j=0; j<P; j++){
48          if (j<L){
49              filtrada[j + 1] = filtrada[j] + (senal[j + Q] - senal[0]) / (2 * L + 1);
50          }
51          else if ((j>=L)&&(j<(P-L))){
52              filtrada[j + 1] = filtrada[j] + (senal[j + Q] - senal[j - L]) / (2 * L + 1);
53          }
54          else{
55              filtrada[j + 1] = filtrada[j] + (senal[P] - senal[j - L]) / (2 * L + 1);
56          }
57      }
58  }
```

- **Negro asignaciones**
- **Naranja sumas y restas**
- **Verde multiplicaciones**
- **Celeste divisiones**

Profilers: valgrind

Medición del uso de recursos

- valgrind --tool=memchech
  - Medición del uso de memoria
- valgrind --tool=callgrind
  - Medición del uso de CPU



03

Entrega 2

Obtención y análisis de resultados

Entrega 2

✦ Ejercicio 1 (Implementación de Filtro en C (no causal) en PC)

El objetivo de este problema es trabajar sobre la implementación realizada en lenguaje C (portable a un sistema embebido) para el práctico anterior. Será realizado en una computadora de uso general de forma de facilitar su desarrollo y verificación.

(a) Realice una estimación teórica del número de operaciones necesarias para implementar el filtro. Realice la estimación, en total, por muestra y por retardo.

(b) Mida el costo computacional utilizando un profiler. Grafique los resultados obtenidos en función del largo de la señal y del largo del filtro, verifique lo esperado teóricamente.

✦ Ejercicio 2 (Implementación de Filtro en C (causal y online) en PC)

El objetivo de este problema es realizar una implementación que sea compatible con su ejecución en tiempo real. Será implementada en una computadora de uso general, para su posterior migración a un sistema embebido.

(a) Modifique el programa del problema anterior para que la función de filtrado reciba solamente una muestra en cada instante de tiempo. Se deberá utilizar solamente el número de retardos necesarios.

(b) Verifique nuevamente el costo computacional. Realice el mismo tipo de verificaciones que en el ejercicio anterior.

Entrega 2

Obtención de resultados

Pasos previos a compilar y analizar con valgrind:

- Comentar los usos de printf()
- Delimitar lo que corresponde al filtro en una única función
  - Para ambos (causal y no causal)
  - No contabilizar los costos de las funciones de lectura/escritura

Entrega 2

Obtención de resultados

¿Bajo qué condiciones son válidos los resultados?

- Cuando se varía una dimensión por vez
- Análisis asintótico
  - Elección cuidadosa de los rangos de L y N

Entrega 2

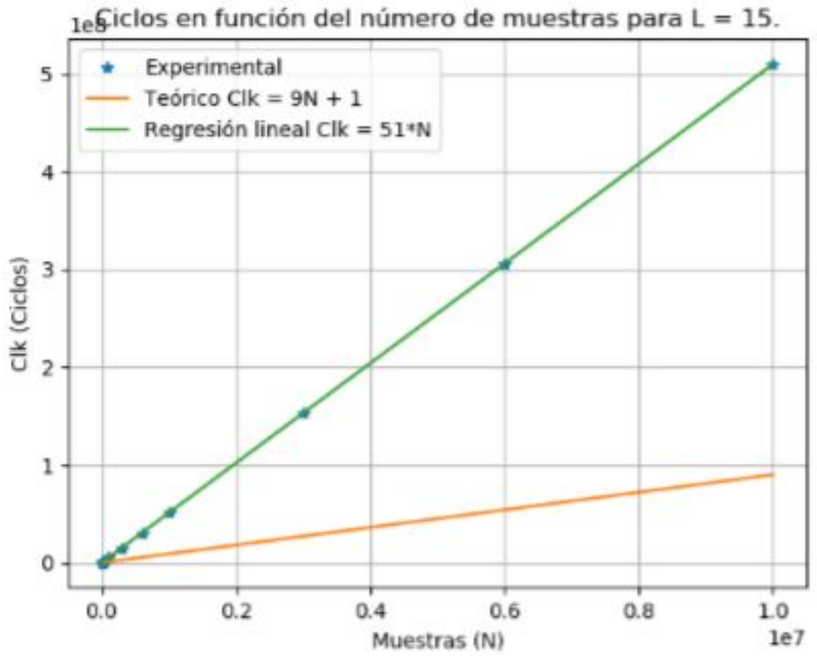
Representación de los resultados

Para su representación en gráficas recordar:

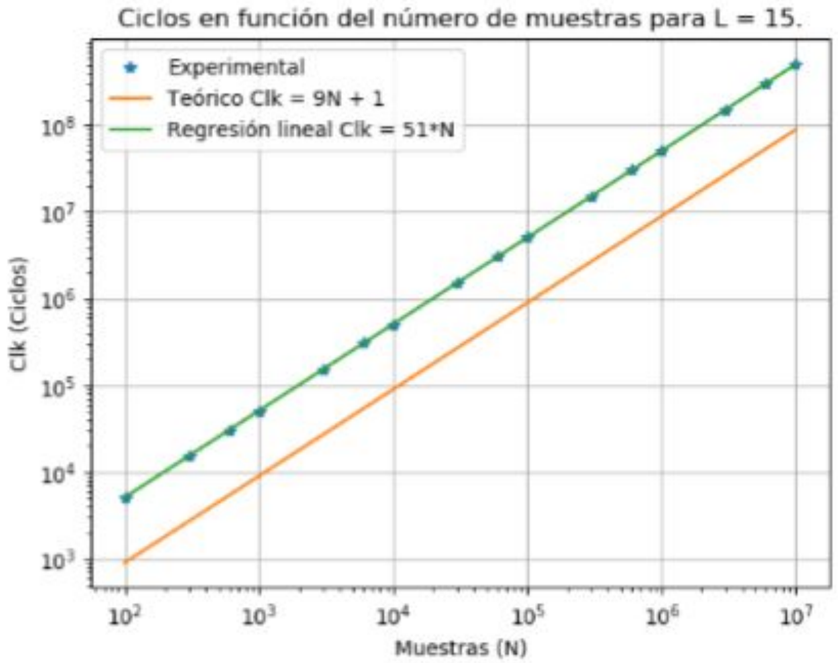
- Utilizar escala logarítmica en AMBOS ejes
- Realizar ajuste de recta y graficarlo

Entrega 2

Representación de los resultados



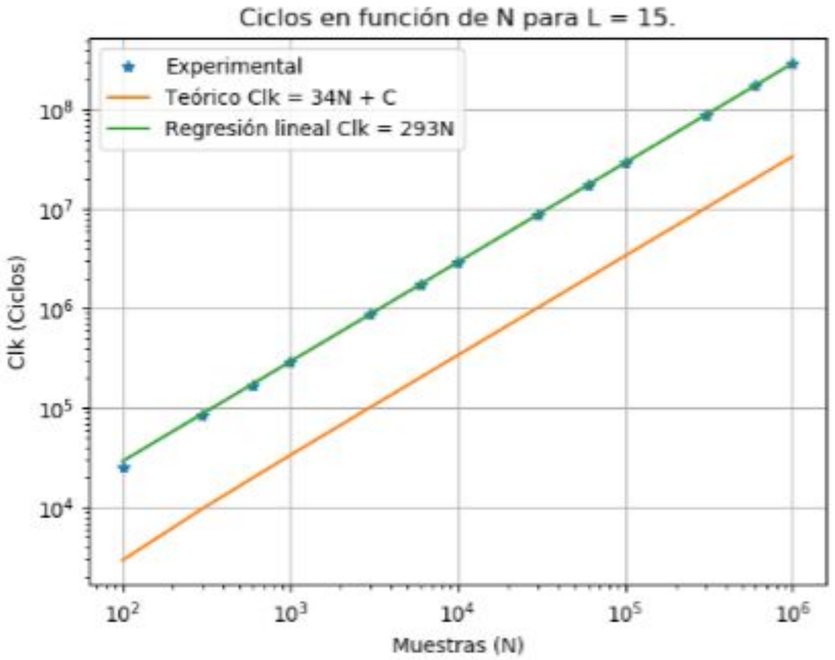
(a) Cálculo de ciclos en función de N para L=15 por distintos métodos.



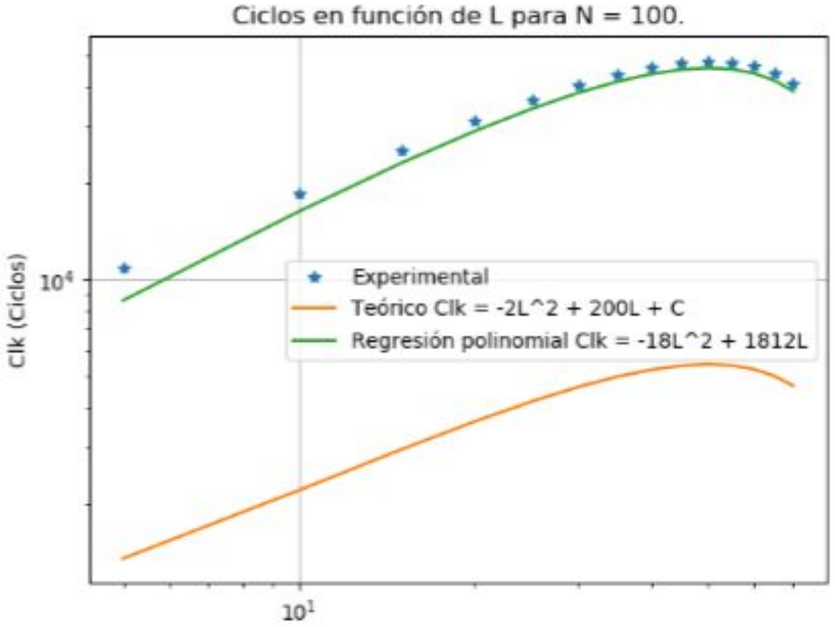
(b) Gráfica en escala logarítmica.

Entrega 2

Representación de los resultados



(a) Calculo de ciclos por distintos métodos en función de N.



(b) Calculo de ciclos por distintos métodos en función de L.

Entrega 2

Análisis de los resultados

- Comparar el ajuste con el comportamiento esperado (teórico)
  - ¿El ajuste es bueno? Valor de R-square
- ¿Qué ocurre antes del main?

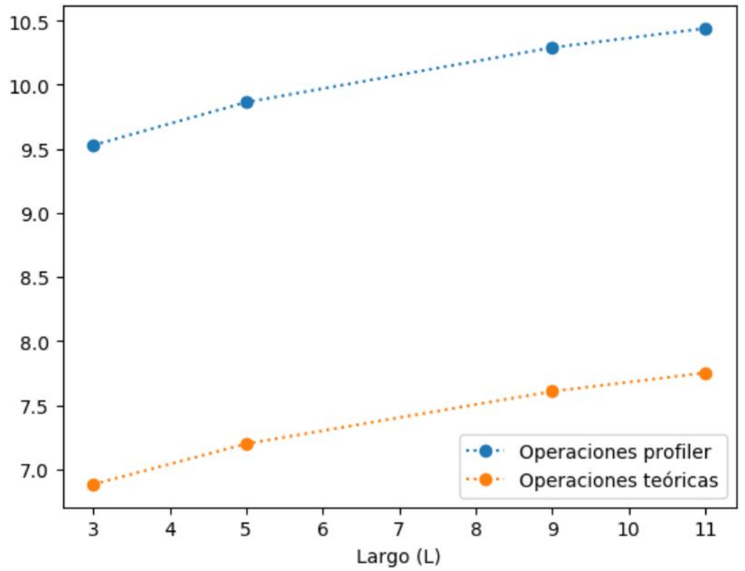


Figura 17: Número de operaciones profiler según tamaño de la ventana del filtro.

Entrega del informe

Viernes 16/09!!!

Gracias

¿Preguntas?

Renato Sosa Machado

renato.sosast@gmail.com

Lucía Lemes

llemes@cup.edu.uy