Introducción al Proyecto

Clase de Práctico N°1
Procesamiento Digital de Señales

INGENIERÍA
BIOLÓGICA

Contenidos

01

Presentación del Proyecto

Señales, bloques y objetivos.

02

Armado del Proyecto

Primer entrega del curso.

03

Repaso de sistemas en TD

Ejemplos: retardo y media móvil.

04

Trabajo práctico

Trabajo en “bloques”. Generación de señales.

01

Presentación del Proyecto

“Tus ojos tienen el control”

Objetivo:

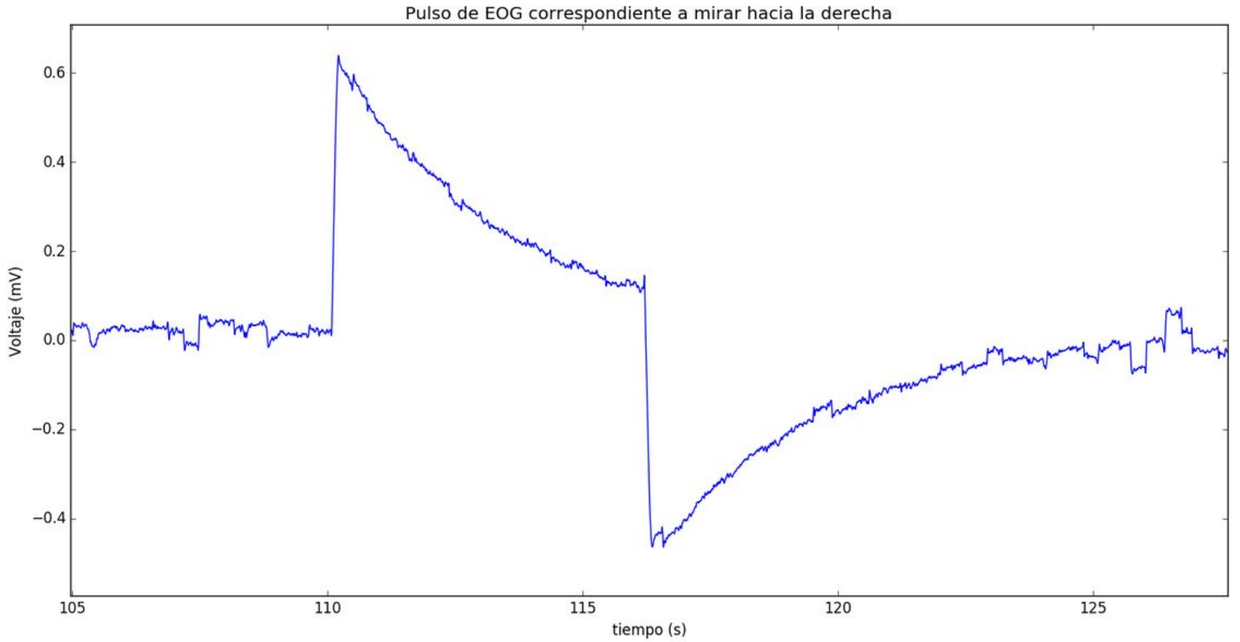
Controlar el movimiento de un objeto mediante señales generadas por nuestro cuerpo.

En este caso, se busca comandar el movimiento de un auto de juguete a partir de señales de Electrooculograma.

Señales

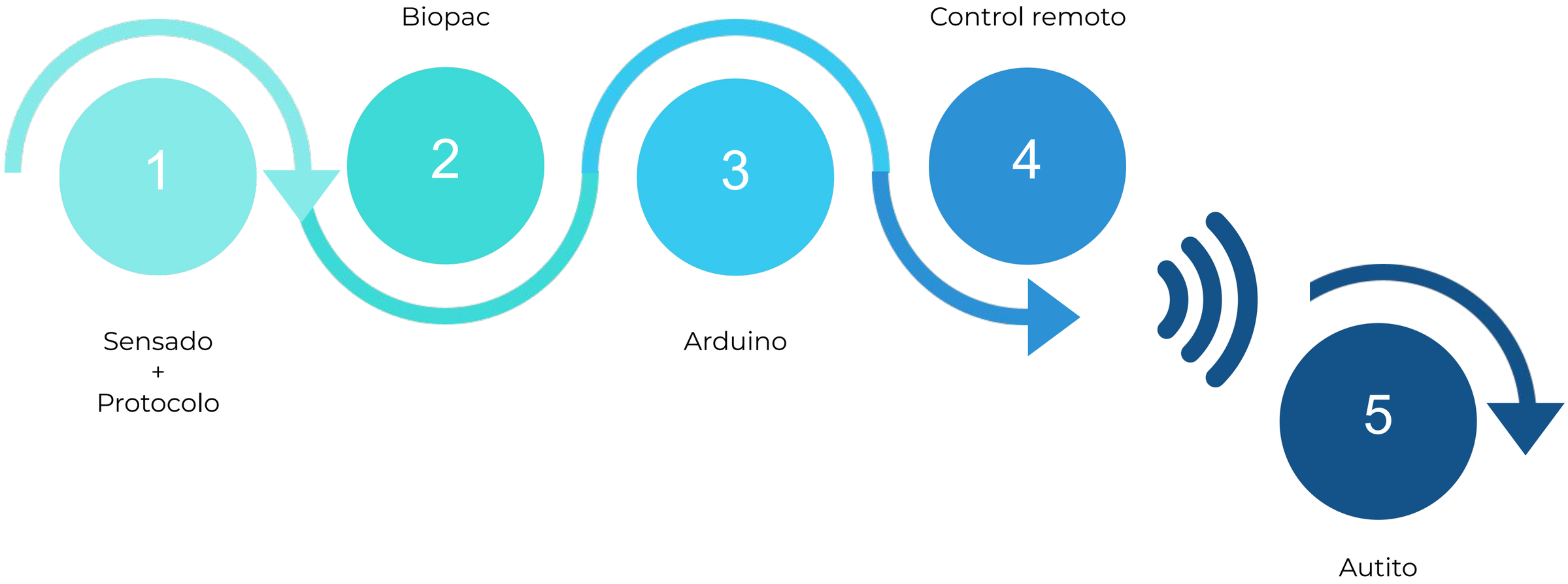
Se emplean señales de Electrooculograma (EOG).

Dependiendo del lado al que se mire (izquierda o derecha) queda determinado el movimiento del autito (avanza o retrocede).



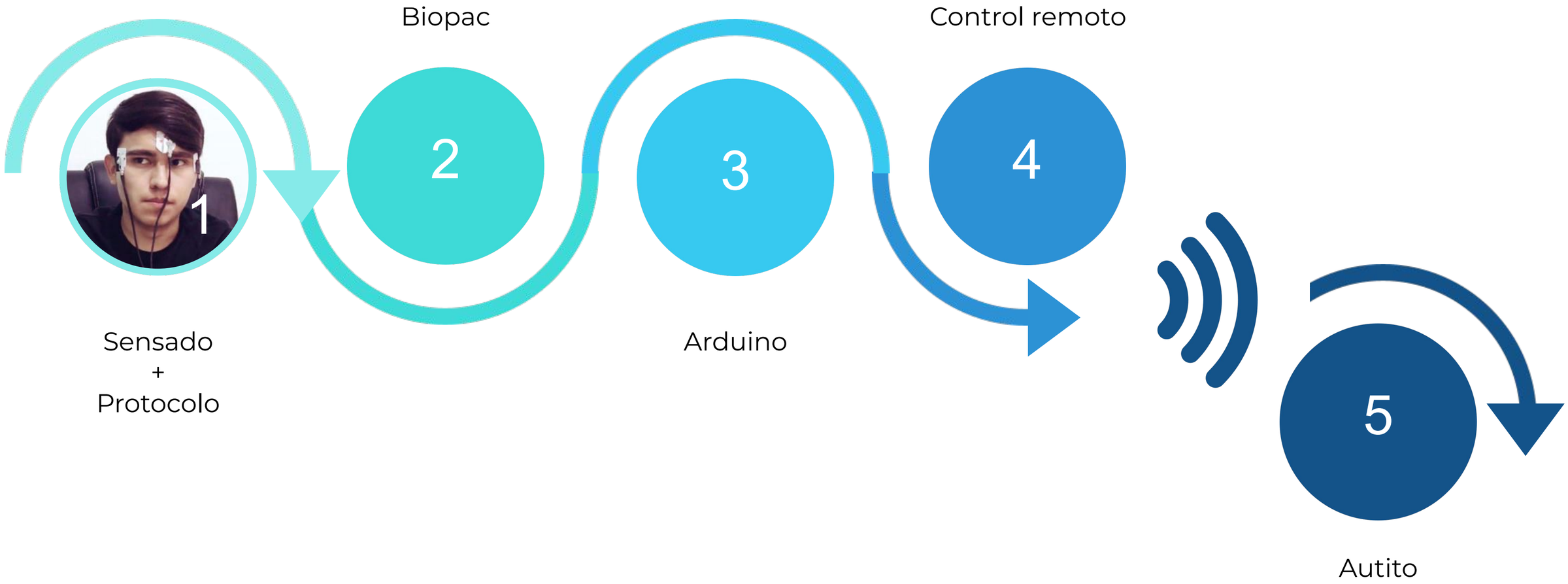
Estudio crítico del autito

Diagrama de bloques



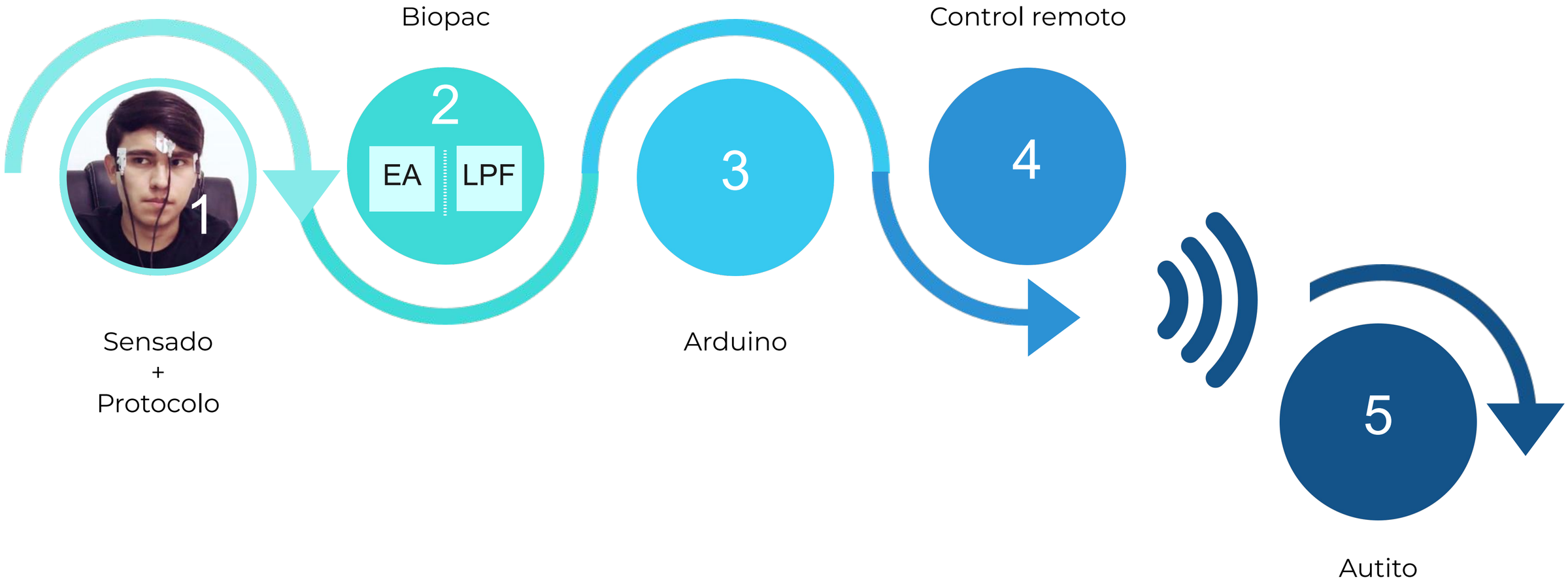
Estudio crítico del autito

Diagrama de bloques



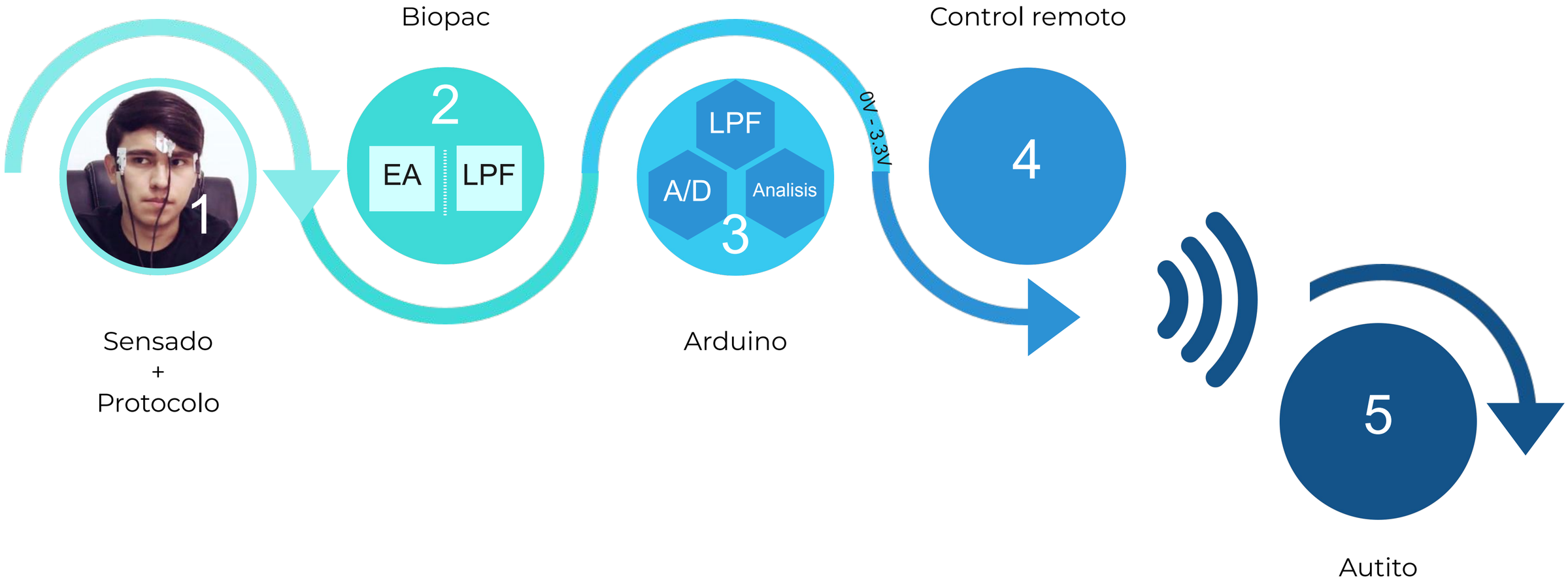
Estudio crítico del autito

Diagrama de bloques



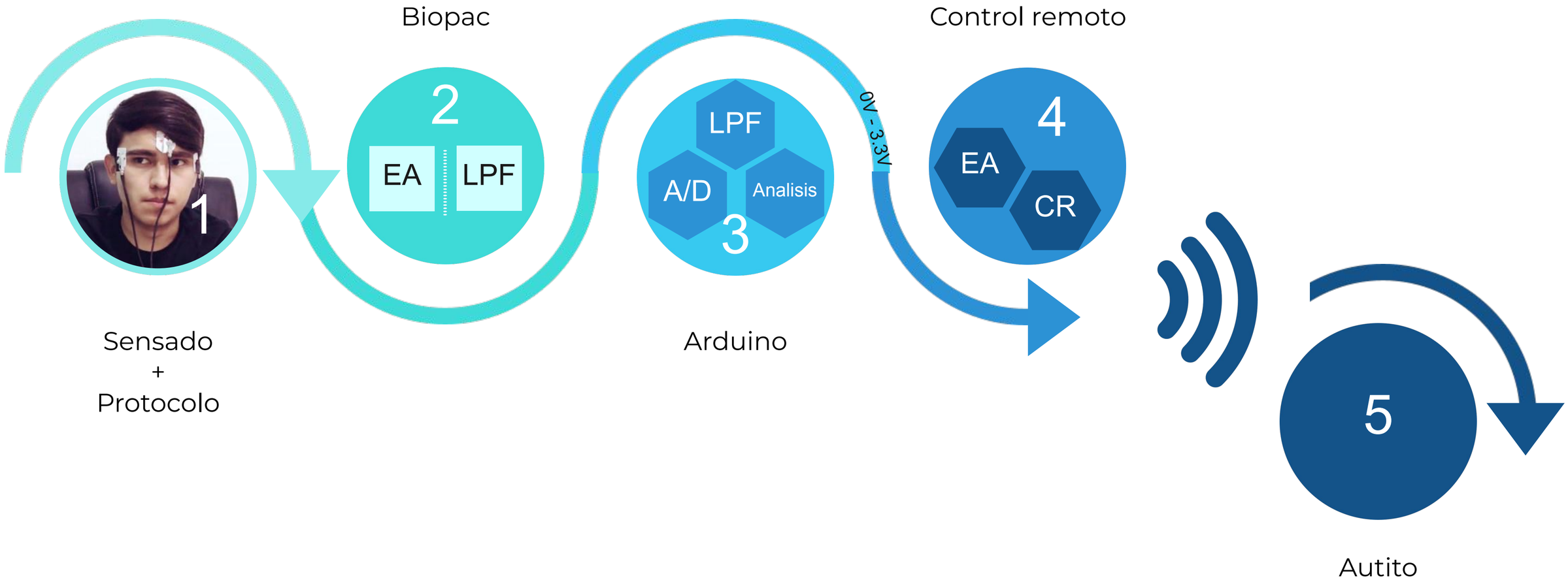
Estudio crítico del autito

Diagrama de bloques



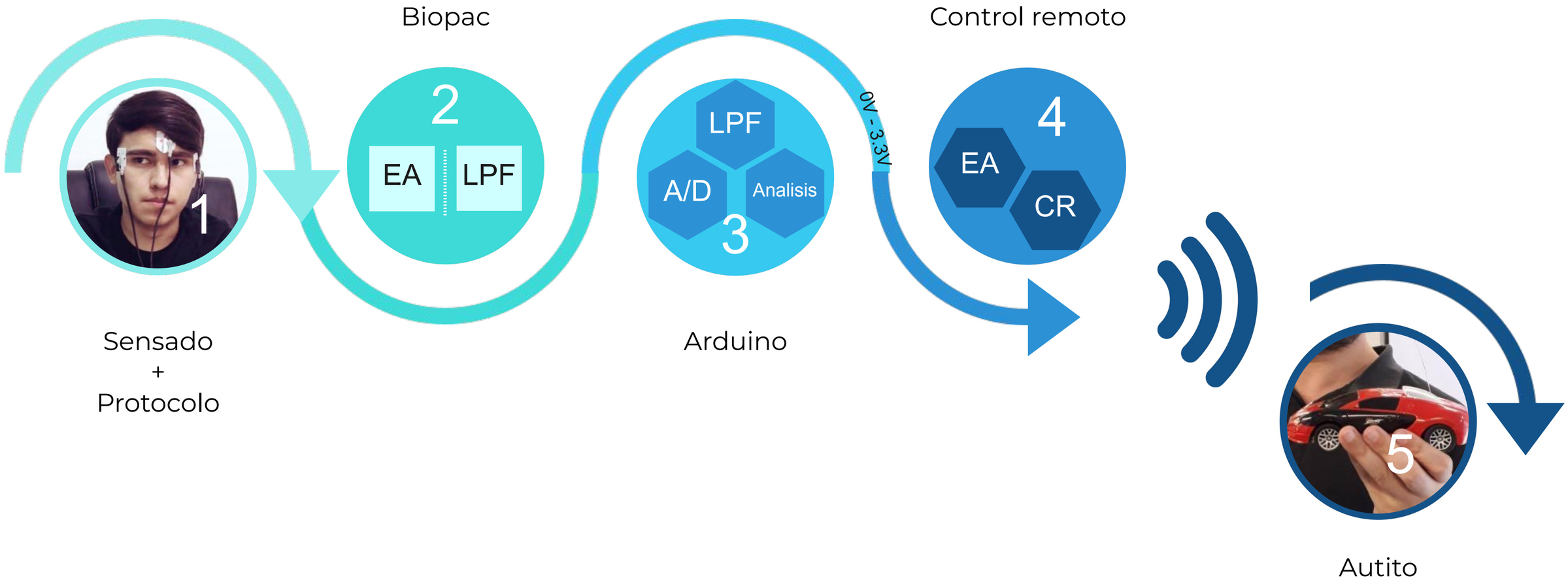
Estudio crítico del autito

Diagrama de bloques



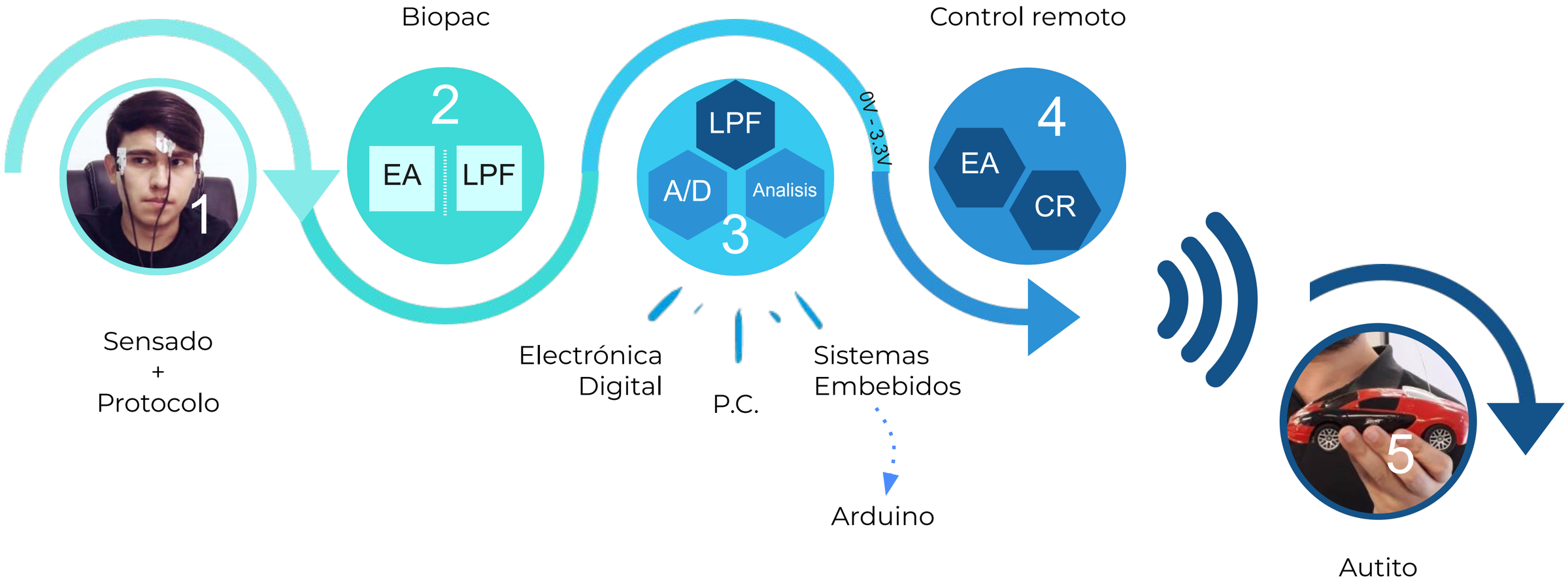
Estudio crítico del autito

Diagrama de bloques



Estudio crítico del autito

Diagrama de bloques



02

Armado del Proyecto

Primera entrega del curso:
Implementación de un filtro FIR

Práctico 01: Implementación en PC

Ejercicio 1 **(Implementación de filtros en un lenguaje de alto nivel (Matlab o Python) en una computadora de alto nivel)**

Para las pruebas iniciales se utilizarán dos señales de entrada: un escalón unitario discreto y una sinusoidal.

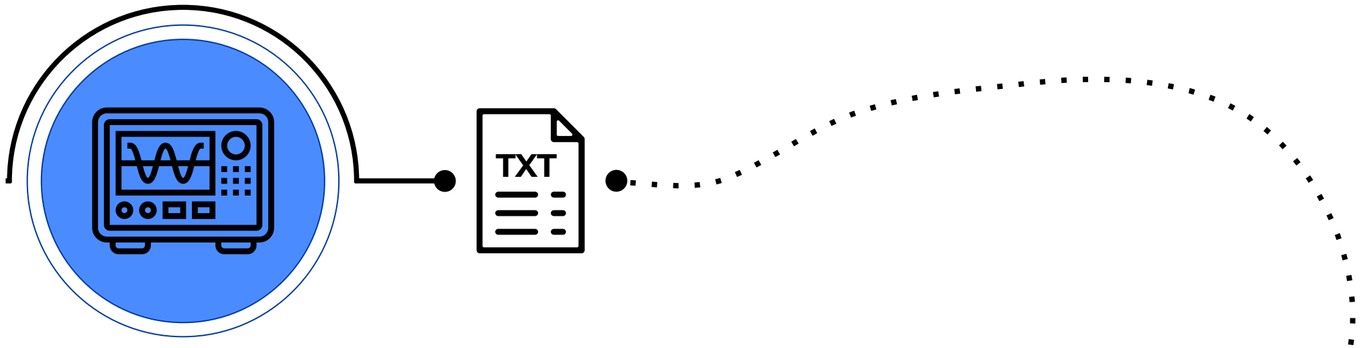
Práctico 01: Implementación en PC

Ejercicio 1 **(Implementación de filtros en un lenguaje de alto nivel (Matlab o Python) en una computadora de alto nivel)**

Para las pruebas iniciales se utilizarán dos señales de entrada: un escalón unitario discreto y una sinusoidal.

(a) Escriba un programa (generar.py) que genere un escalón unitario y lo guarde en un archivo de texto de acuerdo al formato descrito en la Figura 1

Bloque “generar”



Generar

Lenguaje: Python

```
10 #tamaño de la señal.
#a continuación viene la señal, cada muestra separada por un espacio.
0.0 0.0 0.0 0.0 0.0 1.0 1.0 1.0 1.0 1.0
```

Práctico 01: Implementación en PC

Ejercicio 1 **(Implementación de filtros en un lenguaje de alto nivel (Matlab o Python) en una computadora de alto nivel)**

Para las pruebas iniciales se utilizarán dos señales de entrada: un escalón unitario discreto y una sinusoidal.

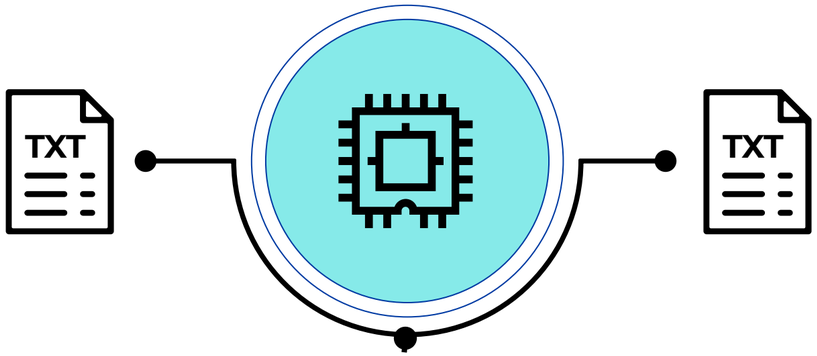
(a) Escriba un programa (generar.py) que genere un escalón unitario y lo guarde en un archivo de texto de acuerdo al formato descrito en la Figura 1

(b) Escriba un programa (procesar.py) que implemente un filtro FIR de forma no causal y con retardo de grupo nulo. Verifique que la respuesta al escalón sea la correcta.

Bloque “procesar”

Procesar

Lenguaje: Python



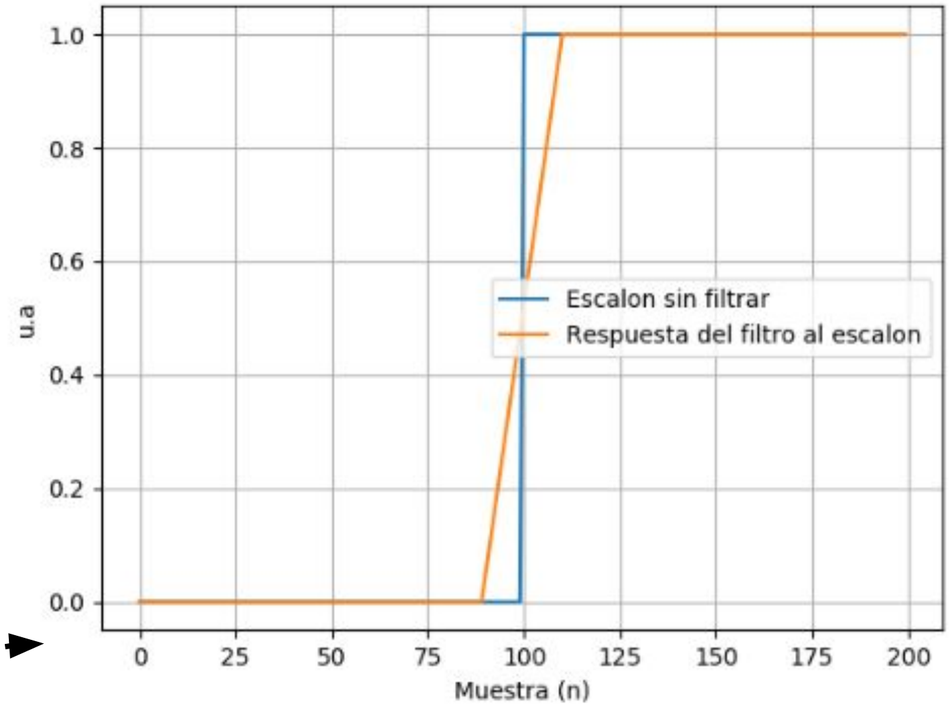


Figura 2: Respuesta del filtro al escalón.

Práctico 01: Implementación en PC

Ejercicio 1 **(Implementación de filtros en un lenguaje de alto nivel (Matlab o Python) en una computadora de alto nivel)**

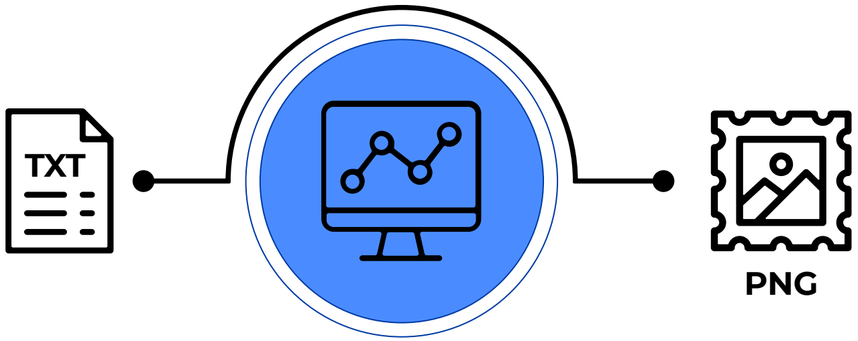
Para las pruebas iniciales se utilizarán dos señales de entrada: un escalón unitario discreto y una sinusoidal.

(a) Escriba un programa (generar.py) que genere un escalón unitario y lo guarde en un archivo de texto de acuerdo al formato descrito en la Figura 1

(b) Escriba un programa (procesar.py) que implemente un filtro FIR de forma no causal y con retardo de grupo nulo. Verifique que la respuesta al escalón sea la correcta.

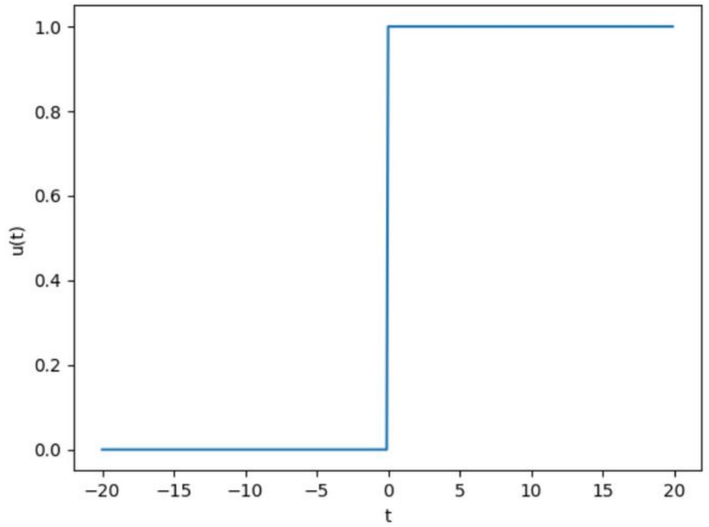
(c) Escriba un programa (visualizar.py) que genere una sinusoidal de amplitud A y frecuencia f y la contamine con ruido aleatorio gaussiano de potencia σ

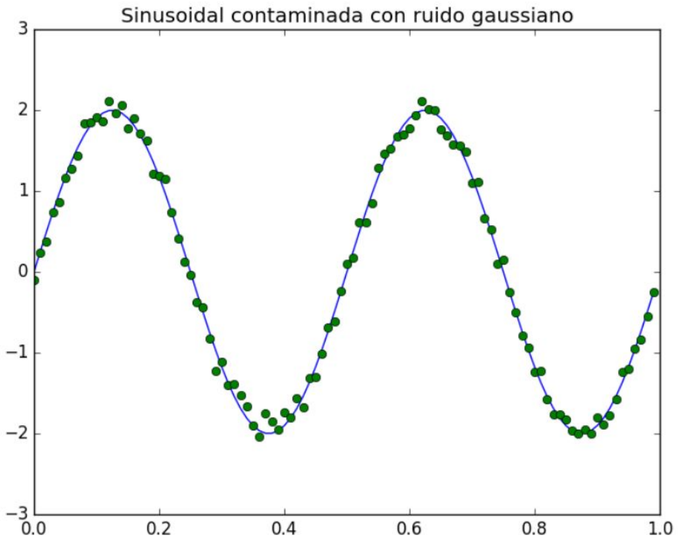
Bloque “visualizar”



Visualizar

Lenguaje: Python





Práctico 01: Implementación en PC

Ejercicio 1 **(Implementación de filtros en un lenguaje de alto nivel (Matlab o Python) en una computadora de alto nivel)**

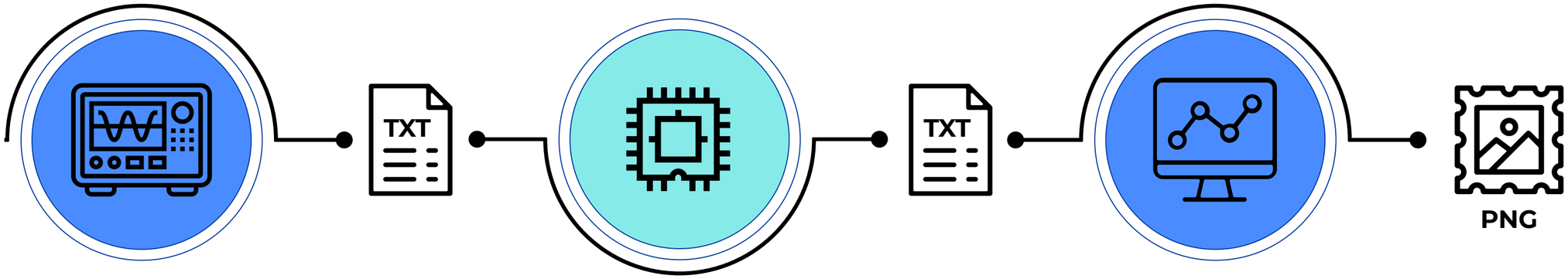
Para las pruebas iniciales se utilizarán dos señales de entrada: un escalón unitario discreto y una sinusoidal.

(a) Escriba un programa (generar.py) que genere un escalón unitario y lo guarde en un archivo de texto de acuerdo al formato descrito en la Figura 1

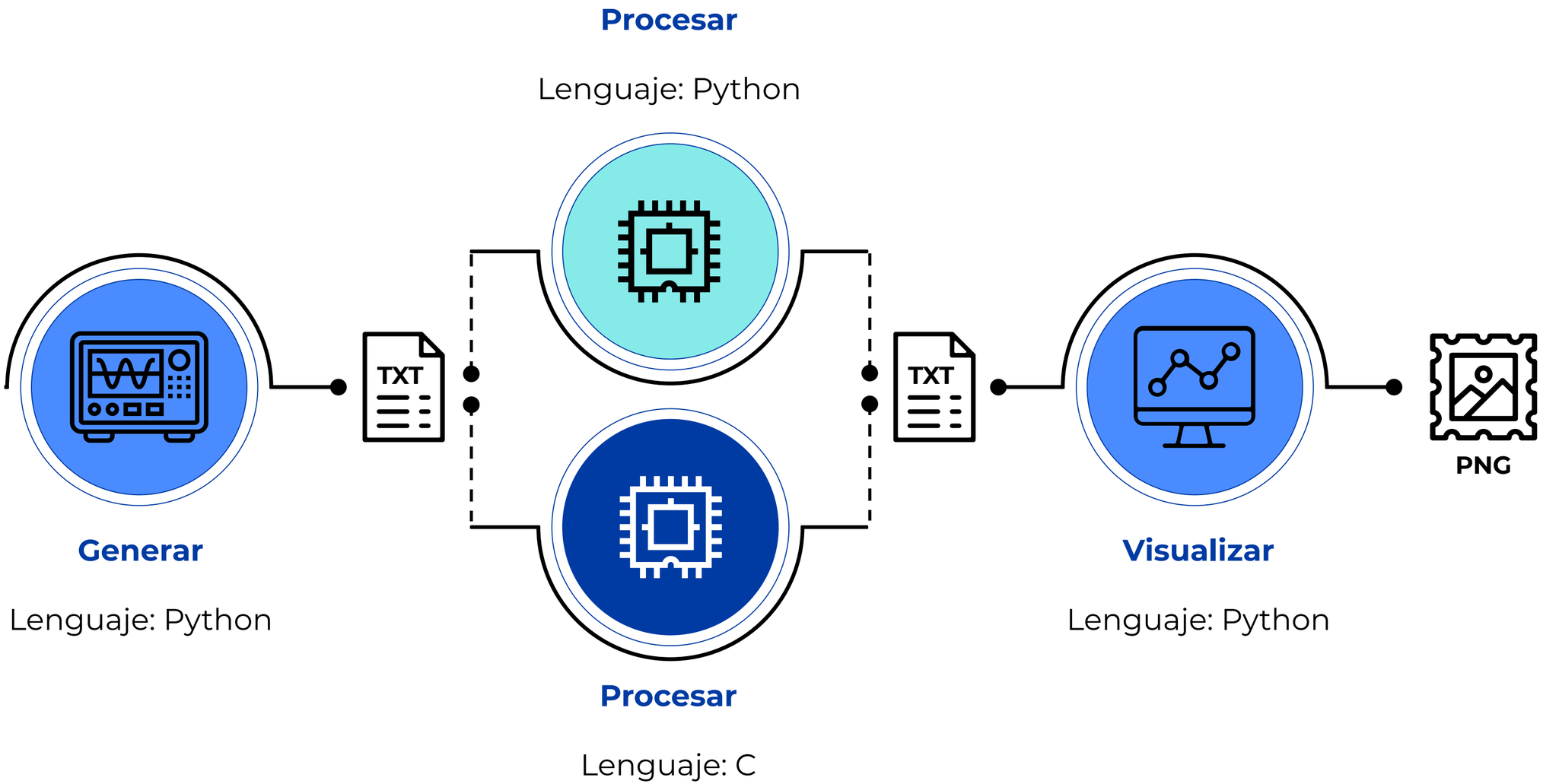
(b) Escriba un programa (procesar.py) que implemente un filtro FIR de forma no causal y con retardo de grupo nulo. Verifique que la respuesta al escalón sea la correcta.

(c) Escriba un programa (visualizar.py) que genere una sinusoidal de amplitud A y frecuencia f y la contamine con ruido aleatorio gaussiano de potencia σ

Trabajo en bloques:



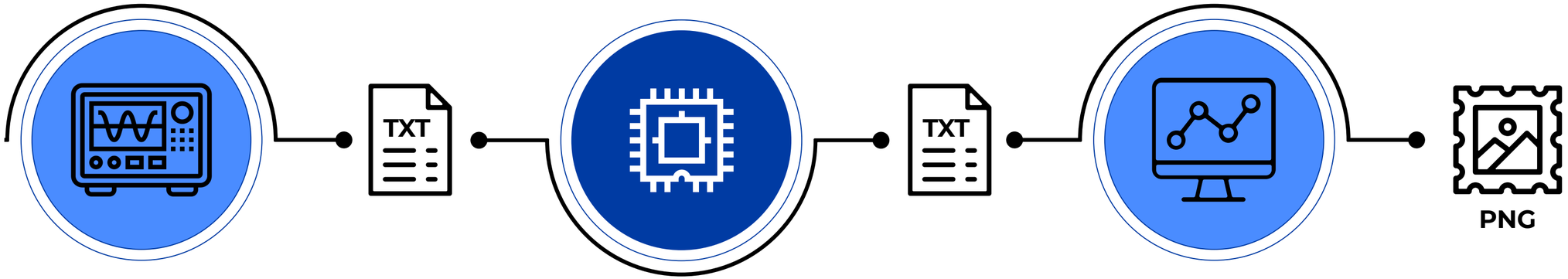
Trabajo en bloques:



Trabajo en bloques:

Procesar

Lenguaje: C



Generar

Lenguaje: Python

Visualizar

Lenguaje: Python

Práctico 01: Implementación en PC

Ejercicio 1 **(Implementación de filtros en un lenguaje de alto nivel (Matlab o Python) en una computadora de alto nivel)**

Para las pruebas iniciales se utilizarán dos señales de entrada: un escalón unitario discreto y una sinusoidal.

(a) Escriba un programa (generar.py) que genere un escalón unitario y lo guarde en un archivo de texto de acuerdo al formato descrito en la Figura 1

(b) Escriba un programa (procesar.py) que implemente un filtro FIR de forma no causal y con retardo de grupo nulo. Verifique que la respuesta al escalón sea la correcta.

(c) Escriba un programa (visualizar.py) que genere una sinusoidal de amplitud A y frecuencia f y la contamine con ruido aleatorio gaussiano de potencia σ

(d) Filtre la señal con una media móvil, comparando señal original con filtrada. Para una primera implementación y a modo de ejemplo utilice los siguientes parámetros: $A = 1$, $f = 100Hz$, $\sigma = 0.1$, $N = 1024$, $L = 5$ y $f_s = 5000Hz$.

(e) Verificar el correcto funcionamiento para diferentes valores del tamaño del filtro. Muestre que el filtro se comporta efectivamente como un pasabajos.

(f) Evalue cuantiativamente el desempeño del filtro. Se utilizará como medida de performance la relación entre la potencia del ruido a la entrada y a la salida. El ruido a la salida se estimará como la diferencia entre la señal original filtrada y la señal con ruido filtrada.

Práctico 01: Implementación en PC

Ejercicio 1 **(Implementación de filtros en un lenguaje de alto nivel (Matlab o Python) en una computadora de alto nivel)**

Para las pruebas iniciales se utilizarán dos señales de entrada: un escalón unitario discreto y una sinusoidal.

(a) Escriba un programa (generar.py) que genere un escalón unitario y lo guarde en un archivo de texto de acuerdo al formato descrito en la Figura 1

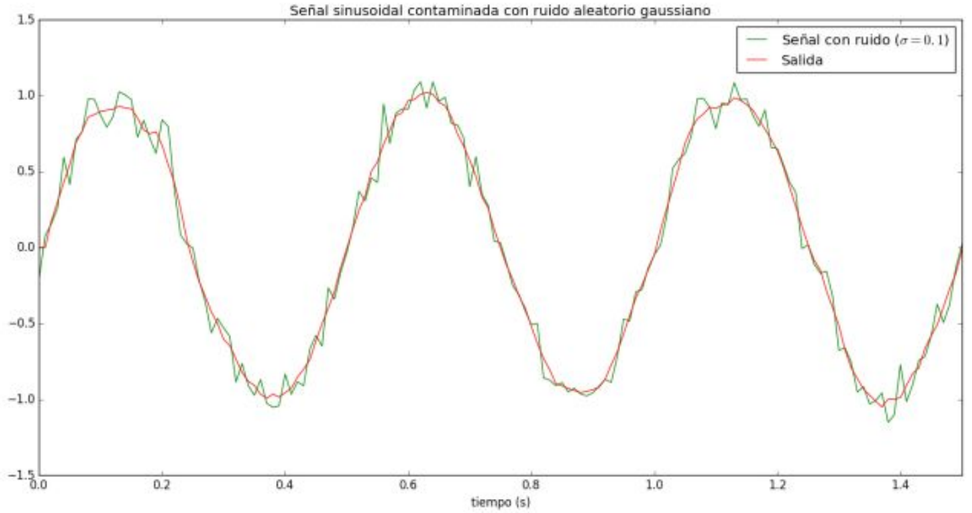
(b) Escriba un programa (procesar.py) que implemente un filtro FIR de forma no causal y con retardo de grupo nulo. Verifique que la respuesta al escalón sea la correcta.

(c) Escriba un programa (visualizar.py) que genere una sinusoidal de amplitud A y frecuencia f y la contamine con ruido aleatorio gaussiano de potencia σ

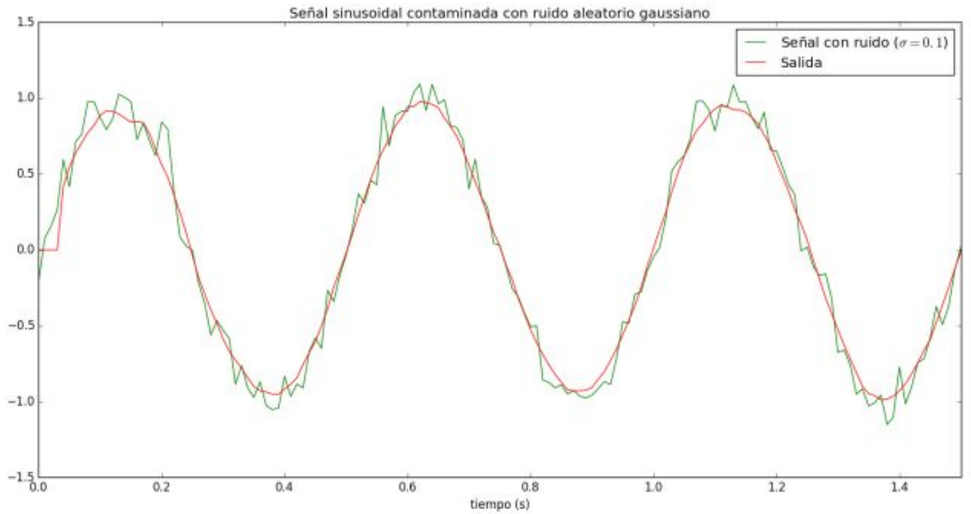
(d) Filtre la señal con una media móvil, comparando señal original con filtrada. Para una primera implementación y a modo de ejemplo utilice los siguientes parámetros: $A = 1$, $f = 100Hz$, $\sigma = 0.1$, $N = 1024$, $L = 5$ y $f_s = 5000Hz$.

(e) Verificar el correcto funcionamiento para diferentes valores del tamaño del filtro. Muestre que el filtro se comporta efectivamente como un pasabajos.

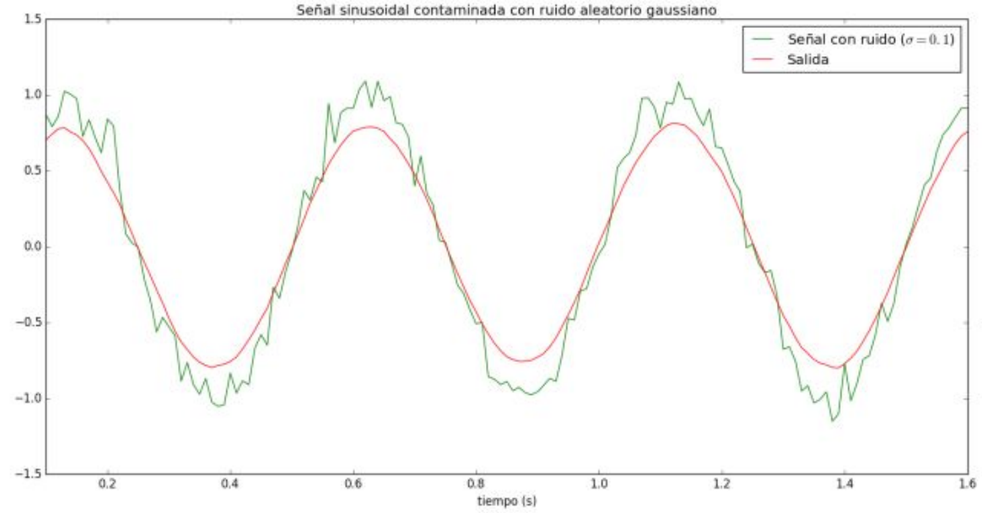
(f) Evalue cuantiativamente el desempeño del filtro. Se utilizará como medida de performance la relación entre la potencia del ruido a la entrada y a la salida. El ruido a la salida se estimará como la diferencia entre la señal original filtrada y la señal con ruido filtrada.



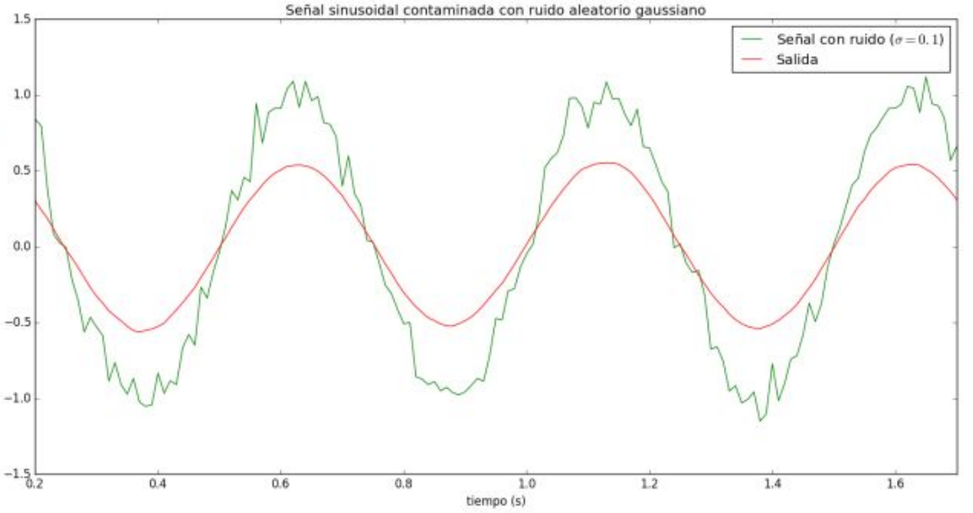
(a) Tamaño de filtro L=5



(b) Tamaño de filtro L=9



(c) Tamaño de filtro L=19



(d) Tamaño de filtro L=29

03

Repaso de sistemas en TD

Retardo y Media Móvil

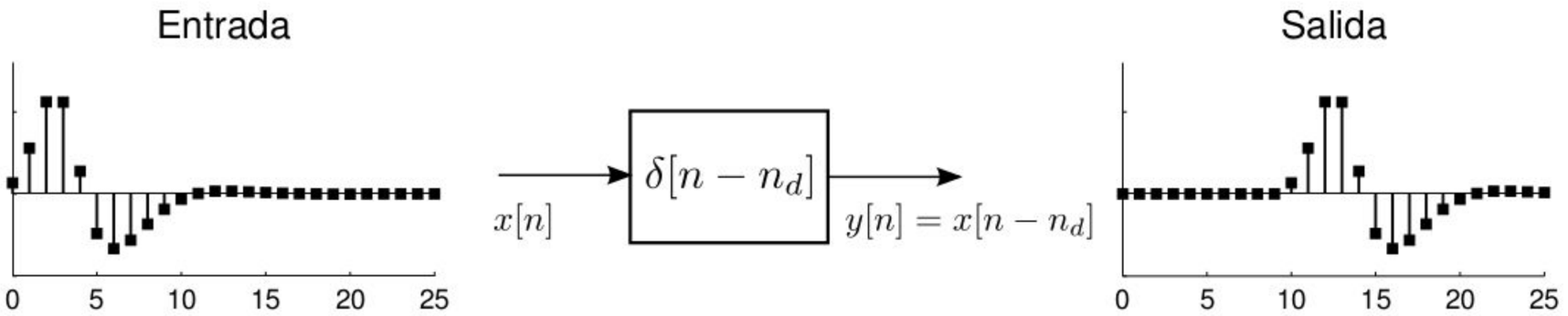
Retardo

- El sistema de retardo se define por la ecuación

$$y[n] = x[n - n_d], \qquad -\infty < n < \infty,$$

donde n_d es un entero fijo que se llama retardo del sistema.

  - El sistema forma la salida desplazando a la secuencia de entrada hacia la derecha una cantidad de n_d muestras (n_d positivo).
  - Si n_d es negativo, el sistema desplaza la secuencia de entrada a la izquierda correspondiendo a un adelanto temporal.

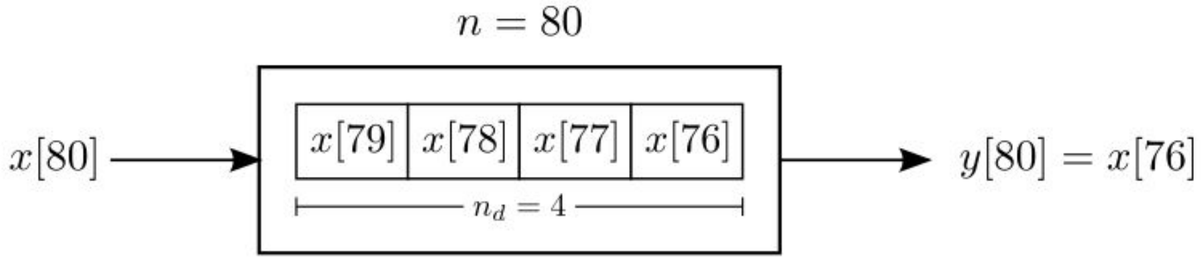


Retardo

- Observaciones:
  - Solo una muestra de la secuencia de entrada es usada para calcular una muestra de la secuencia de salida. Por ejemplo, si $n_d = 4$:

$$y[80] = x[76], \quad y[81] = x[77], \quad y[82] = x[78], \quad \ldots$$

  - El sistema necesita almacenar n_d muestras de la entrada. Si $n_d = 4$, en $n = 80$ se necesita tener almacenado $x[76]$, $x[77]$, $x[78]$, $x[79]$, para dar la salida en $n = 80, 81, 82, 83$:



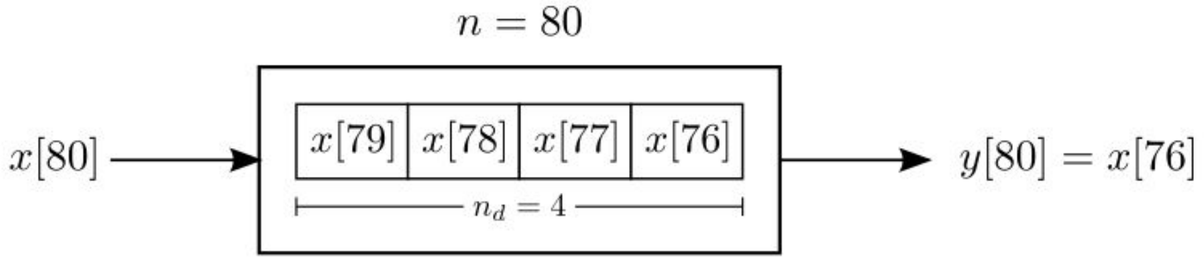
  - Si $n_d < 0$, para calcular la salida se necesitan muestras futuras de la entrada. Para eso, el sistema debería poder predecir el futuro.

Retardo

- Observaciones:
  - Solo una muestra de la secuencia de entrada es usada para calcular una muestra de la secuencia de salida. Por ejemplo, si $n_d = 4$:

$$y[80] = x[76], \quad y[81] = x[77], \quad y[82] = x[78], \quad \ldots$$

  - El sistema necesita almacenar n_d muestras de la entrada. Si $n_d = 4$, en $n = 80$ se necesita tener almacenado $x[76]$, $x[77]$, $x[78]$, $x[79]$, para dar la salida en $n = 80, 81, 82, 83$:



  - Si $n_d < 0$, para calcular la salida se necesitan muestras futuras de la entrada. Para eso, el sistema debería poder predecir el futuro.

Media Móvil

- El sistema de Media Móvil Causal se define por la ecuación

$$y[n] = \frac{1}{M+1}\sum_{k=0}^{M} x[n-k]$$

$$= \frac{1}{M+1}\{x[n] + x[n-1] + x[n-2] + \cdots + x[n-M]\}$$

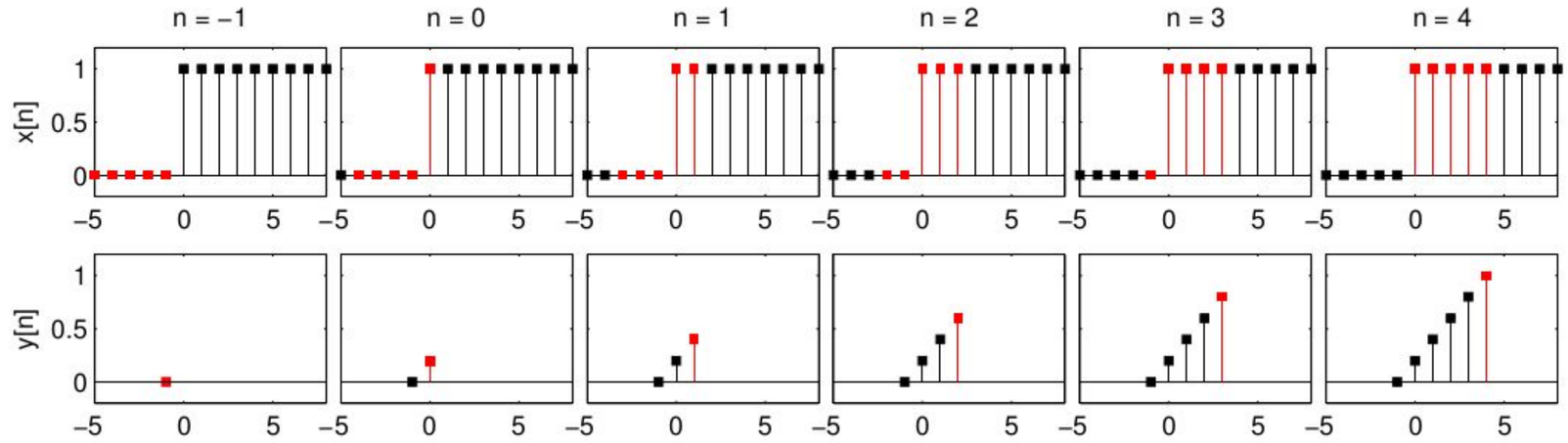
- ¿Qué hace el sistema? Considérese la salida en $n = 80$ si $M = 4$:

$$y[80] = \frac{x[80] + x[79] + x[78] + x[77] + x[76]}{5}$$

  - El sistema calcula la salida como el promedio de la últimas $M+1$ muestras de la entrada.
- Observación:
  - La muestra actual de la salida es función de la muestra actual y M muestras previas de la entrada.
  - El sistema tiene que poder almacenar las M muestras previas de la entrada.

Media Móvil

- El nombre del filtro proviene de que la salida es la media de la señal en una ventana deslizante.



- Implementación como sistema recursivo
  Es posible calcular la salida $y[n]$ realizando menos operaciones si se usa el valor anterior de la salida en el tiempo $n-1$:

$$y[80] = \frac{x[80]+x[79]+x[78]+x[77]+x[76]}{5}$$

$$y[81] = \frac{x[81]+x[80]+x[79]+x[78]+x[77]}{5}$$

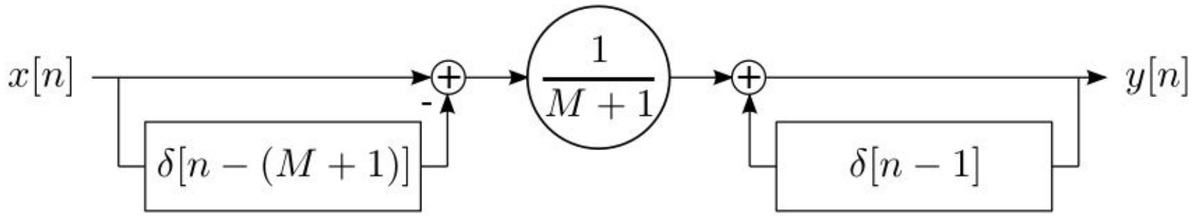
$$y[81] = y[80] + \frac{x[81]-x[76]}{5}$$

Media Móvil

- Implementación como sistema recursivo: La ecuación genérica del sistema de media móvil causal es

$$y[n] = y[n-1] + \frac{x[n] - x[n-(M+1)]}{M+1}$$

- Se necesitan solo tres operaciones para calcular la salida.
- Se necesita almacenar solo dos valores, $y[n-1]$ y $x[n-(M+1)]$



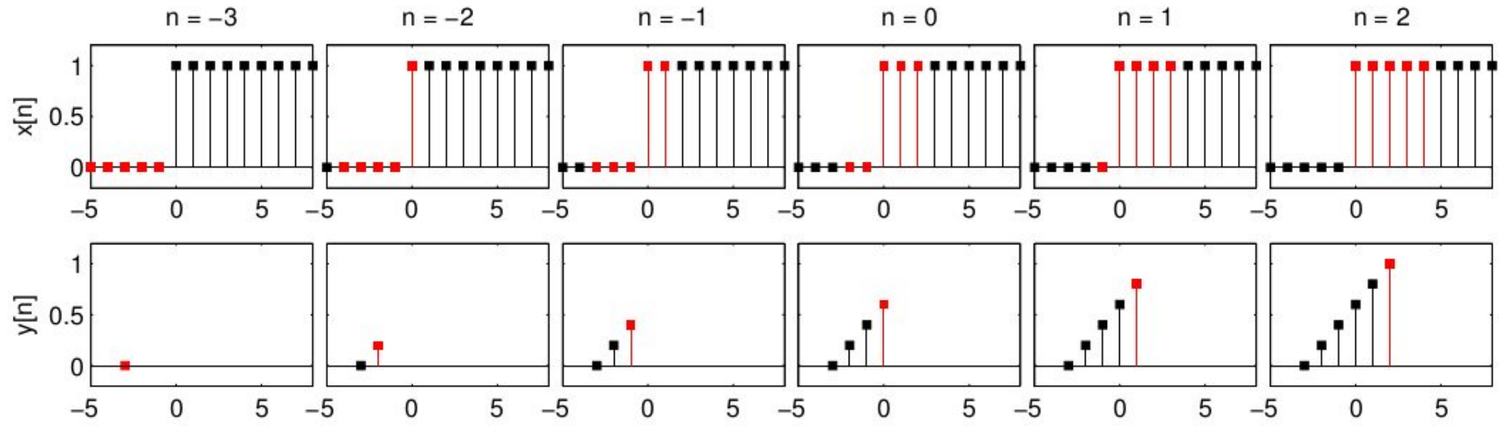
- La forma general del filtro de media móvil es

$$\begin{aligned} y[n] &= \frac{1}{M_1 + M_2 + 1} \sum_{k=-M_1}^{M_2} x[n-k] \\ &= \frac{1}{M_1 + M_2 + 1} \{x[n+M_1] + x[n+M_1-1] + \cdots + x[n] \\ &\quad + x[n-1] + \cdots + x[n-M_2]\} \end{aligned}$$

Media Móvil

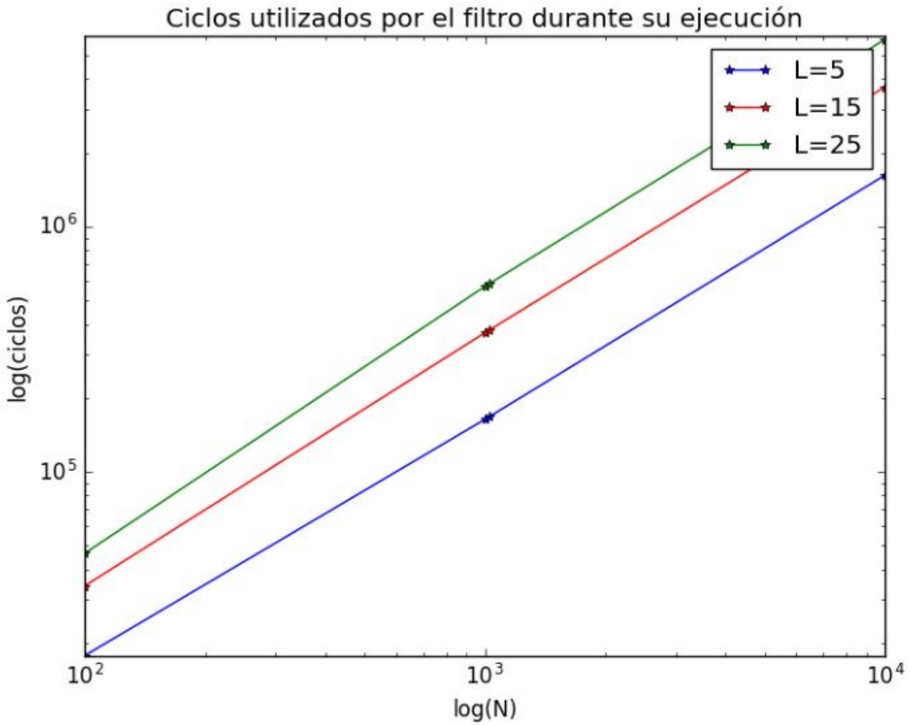
- Considérese la salida en $n = 80$ si $M_1 = 2$ y $M_2 = 2$

$$y[80] = \frac{x[82] + x[81] + x[80] + x[79] + x[78]}{5}$$

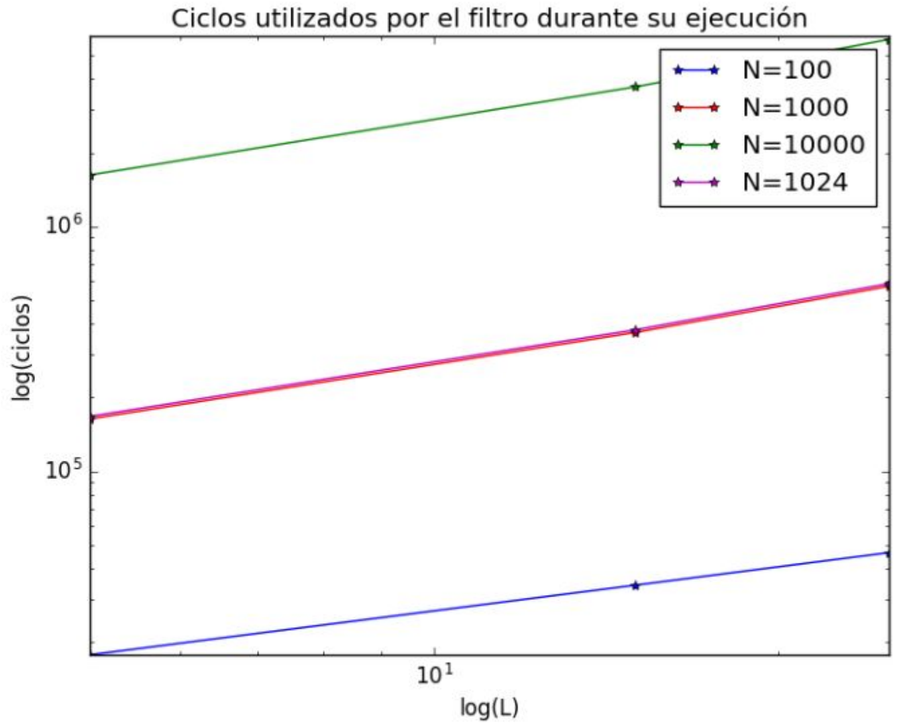


- Observaciones:
  - En el filtro de media móvil general, la salida depende de muestras futuras de la entrada.
  - Para que la salida no dependa de muestras futuras de la entrada se tiene que cumplir que $-M_1 \geq 0$ y $M_2 \geq 0$.

Media Móvil: Complejidad



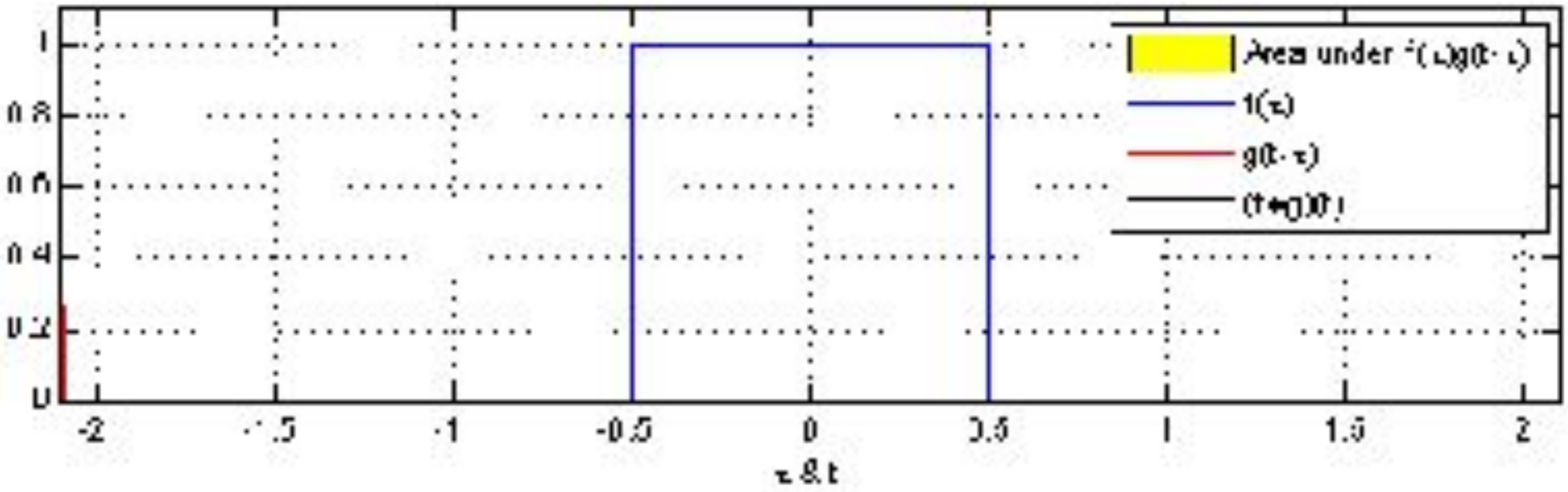
(a) Cantidad de ciclos realizados en función de la cantidad de muestras



(b) Cantidad de ciclos realizados en función del tamaño del filtro

Figura 7: Relación entre ciclos y muestras, y entre ciclos y largo del filtro FIR

Media Móvil: Entrada de pulso



04

Trabajo sobre el práctico

Implementación de los bloques y repaso de entornos virtuales

Tareas:

1. Crear los tres scripts requeridos:
   - generar
   - procesar
   - visualizar

2. Iniciar la implementación del bloque “generar”.

   Trabajar utilizando entornos virtuales

Creación de un entorno virtual

- Instalar venv que permite crear entornos virtuales:
  - `sudo apt-get install python3-venv`
- Crear el entorno:
  - `python3 -m venv “nombre del entorno”`
  - Se crea el entorno con el nombre dado y se además se crea una carpeta en el directorio donde se almacenarán los datos del proyecto.

Creación de un entorno virtual en Anaconda

- `conda create -n nombreenv python=x.x`
  - Donde “nombreenv” es el nombre que quieres dar a tu entorno y “x.x” la versión de Python que quieres dar a tu entorno.
- `source activate nombreenv`
  - Si se ha activado correctamente a la derecha de nuestro terminal debe salir entre paréntesis el nombre de nuestro entorno.
- Los paquetes se puede instalar con **conda install** o **pip install**

Gracias

¿Preguntas?

Renato Sosa Machado

renato.sosast@gmail.com

Lucía Lemes

llemes@cup.edu.uy