

PROYECTO 4

CLASIFICACIÓN DE SEÑALES ECG



Isabela Trujillo Betancourt

Laura Esperanza Vargas Patarroyo

¿QUÉ ES NEUROKIT2?



Es un paquete de Python de código abierto que proporciona un acceso sencillo a rutinas avanzadas de procesamiento de bioseñales permitiendo analizar señales fisiológicas y datos neurofisiológicos [1].

Incluye herramientas para procesar una variedad de señales corporales, como: ECG, PPG, EDA, EMG, RSP.

Herramientas para pasos de procesamiento específicos

Extracción de tasas y métodos de filtrado

Procesar datos en pocas líneas de código

Control preciso sobre los argumentos y parámetros

INSTALACIÓN E IMPORTACIONES

```
!pip install neurokit2
import neurokit2 as nk
import matplotlib.pyplot as plt
```

1

Es necesario realizar la instalación del paquete e importar el módulo

NOTA:
Es importante importar todo aquello que necesites adicionalmente para que tu script funcione

SIMULACIÓN ECG

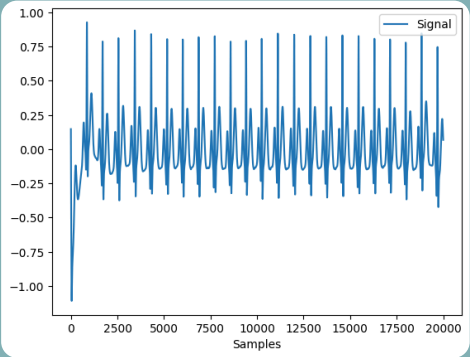
```
# Simulación ECG
ecg = nk.ecg_simulate(duration=10, sampling_rate=1000, heart_rate=80)
# Visualizar
nk.signal_plot(ecg)
```

2



ecg_simulate
Genere una señal de ECG artificial de una duración y frecuencia de muestreo determinadas

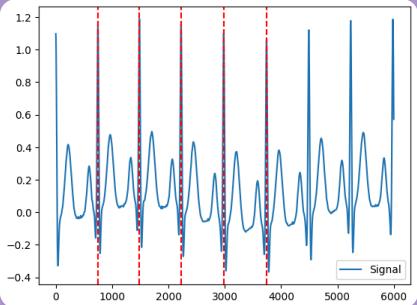
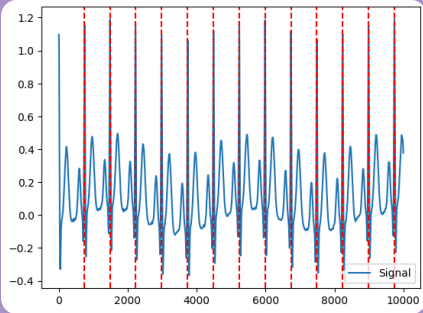
signal_plot
Permite visualizar las señales



ECG_R_PEAKE

Los picos R son marcados como "1" en una lista de ceros

```
# Visualización Picos R
_, rpeaks = nk.ecg_peaks(ecg, sampling_rate=1000)
nk.events_plot(rpeaks['ECG_R_Peaks'], ecg)
```



```
# Acercamiento de los primeros 5 Picos R
plot = nk.events_plot(rpeaks['ECG_R_Peaks'][:5], ecg[:6000])
```

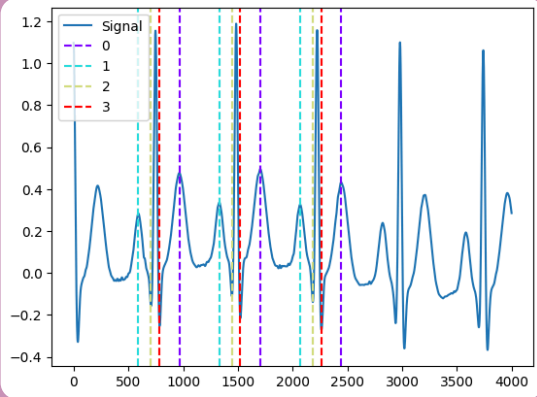
PEAKS

```
# Limitar el Ciclo Cardiaco
_, waves_peak = nk.ecg_delineate(ecg, rpeaks, sampling_rate=1000)
# Acercamiento a los picos PQST
nk.events_plot([waves_peak['ECG_T_Peaks'][:3],
               waves_peak['ECG_P_Peaks'][:3],
               waves_peak['ECG_Q_Peaks'][:3],
               waves_peak['ECG_S_Peaks'][:3]],
               ecg[:4000])
```

ECG_T_Peaks
ECG_P_Peaks
ECG_Q_Peaks
ECG_S_Peaks

Los picos respectivos son marcados como "1" en una lista de ceros

ecg_delineate
Permite delinear el complejo QRS, es decir, las diferentes ondas de los ciclos cardiacos

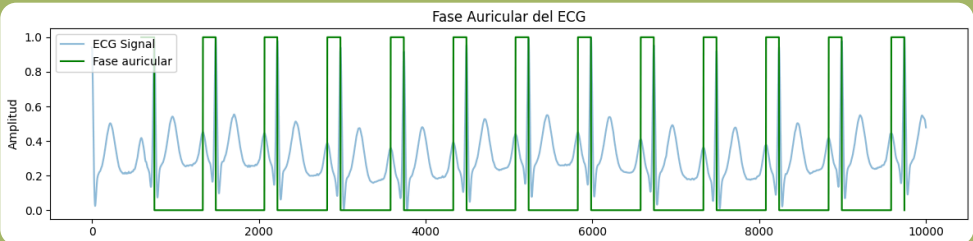


ECG_PHASE_ATRIAL

```
cardiac_phase = nk.ecg_phase(ecg, rpeaks, delineate_info=waves_peak, sampling_rate=1000)

plt.figure(figsize=(12, 6))
plt.subplot(2, 1, 1)
plt.plot(nk.rescale(ecg), label="ECG Signal", alpha=0.5)
plt.plot(cardiac_phase["ECG_Phase_Atrial"], label="Fase auricular", color="green")
plt.title("Fase Auricular del ECG")
plt.ylabel("Amplitud")
plt.legend()
```

Fase cardíaca, marcada por "1" para la sístole y "0" para la diástole



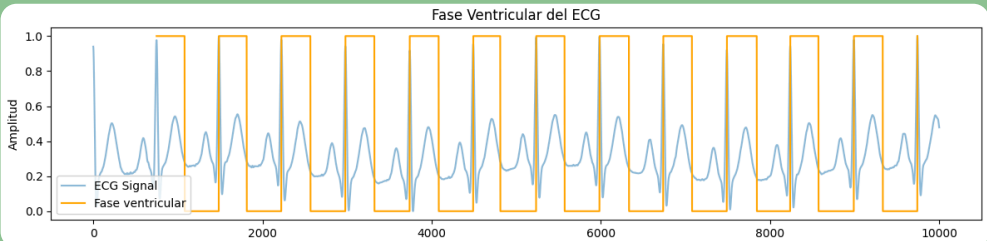
ecg_phase

Calcular la fase cardíaca (tanto auricular como ventricular)

ECG_PHASE_VENTRICULAR

```
plt.subplot(2, 1, 2)
plt.plot(nk.rescale(ecg), label="ECG Signal", alpha=0.5)
plt.plot(cardiac_phase["ECG_Phase_Ventricular"], label="Fase ventricular", color="orange")
plt.title("Fase Ventricular del ECG")
plt.ylabel("Amplitud")
plt.legend()
```

Indicación de si el inicio del evento coincide con la sístole respiratoria (1) o diástole (0)



nk.rescale

Normaliza los valores de una señal para que estén dentro de un rango deseado

BIBLIOGRAFÍA

[1] Dominique, Makowski., Tam, Pham., Zen, J., Lau., Jan, C., Brammer., François, Lespinasse., Hung, Pham., Christopher, Schölzel., S., H., Annabel, Chen. (2021). NeuroKit2: A Python toolbox for neurophysiological signal processing. Behavior Research Methods, 53(4):1689-1696. doi: 10.3758/S13428-020-01516-Y