

# Ejercicio

## - Gráficos con Python

Empleando la librería *matplotlib.pyplot*, se desarrolla una función que permite generar diferentes estructuras de gráficos.

## - Ejercicio

- Generar una función en Python que permita graficar sub-plots con diferentes distribuciones.

La función debe generar múltiples sub-plots en una sola figura según los siguientes parámetros:

**figPlot**(x, y, row, col, joinVec, numplot, show=False, typeGraf, xlim, ylim, xlabel, ylabel)

- **x, y** : Rango de valores a graficar. Se ingresan tantos vectores como gráficos a realizar.
  - **row, col** : Filas y columnas de la figura.
  - **joinVec** : Vector que especifica como se agrupan los gráficos. Ej. row,col: 3,3. joinVec:(1,3),4,5,6,(7,9).
  - **numplot** : Nombre del plot en número, el cual se incluye en figure(numplot).
  - **show** : Habilita el gráfico.
  - **typeGraf** : Selecciona el tipo de gráfico en plot o stem y se deben ingresar tantos tipos como gráficos a realizar.
  - **xlim, ylim** : Límites de los ejes y se deben pasar tantos límites como gráficos a realizar.
  - **xlabel, ylabel** : Nombre de los ejes X e Y pero solamente se los asignaremos a los gráficos de los márgenes izquierdo e inferior.
- Generar otra función **graficar()** que haga de interfaz entre el usuario y la función figPlot(), en la cual se debe tener en cuenta que:
    - Se debe corroborar la coherencia entre la cantidad de sub-plots y la distribución de los mismos, en caso contrario se deberá volver a pedir los datos. Asumir que sólo se pueden agrupar sub-plots horizontalmente.
    - Se debe generar con algún criterio los vectores de datos para los distintos gráficos (combinar distintas funciones con el uso de random).
    - Los datos que ingresa el usuario son row, col, joinVec, numplot, show, typeGraf, xlim, ylim, xlabel e ylabel.