

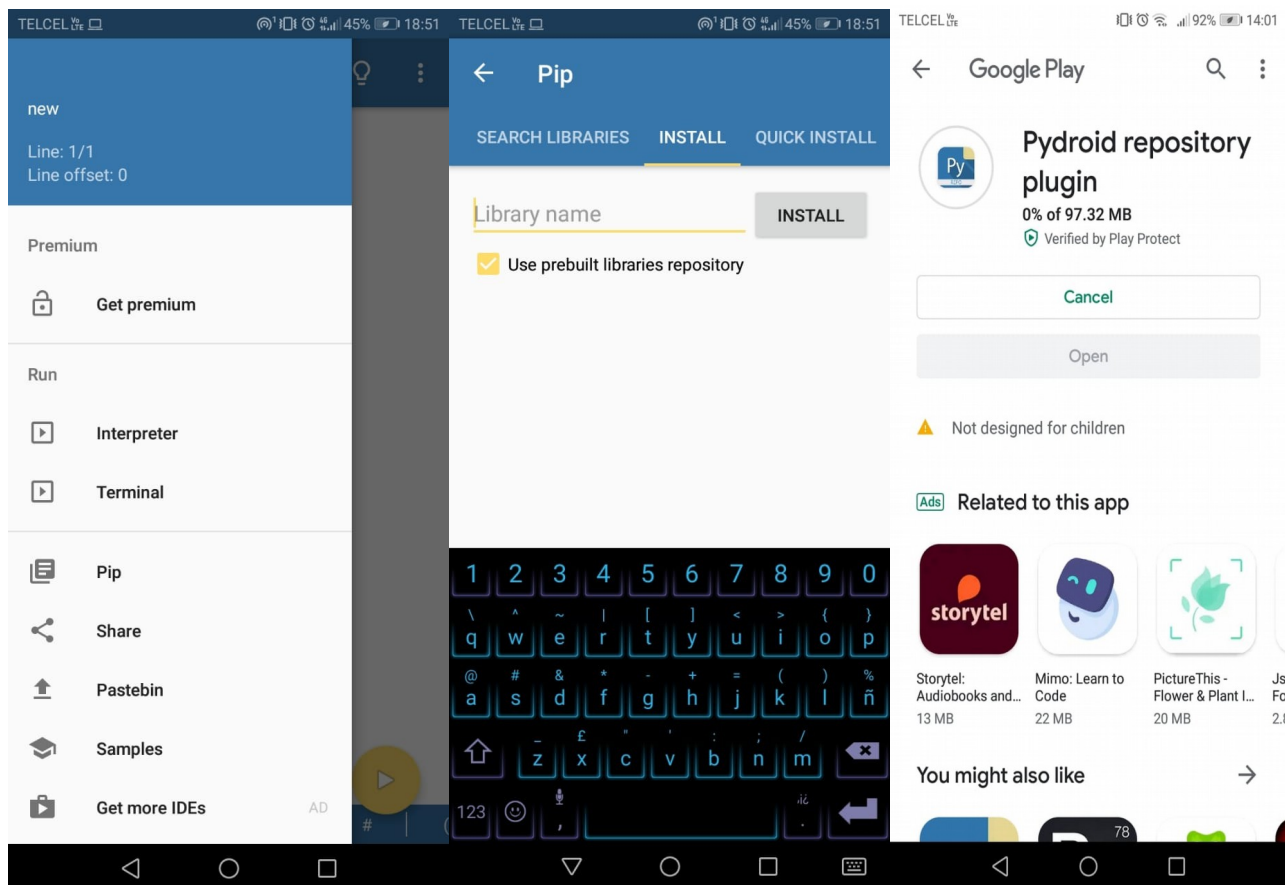
Como instalar bibliotecas necesarias

1. En la parte de menu (las tres rayitas) encontrar la opción Pip y seleccionarlo
2. Al estar adentro, teclear los nombres de las bibliotecas (una por una) y a continuación presionar install (instalar)

Instalar:

- numpy

nota: Es necesario descargar el repositorio de la play store puesto que la biblioteca no viene predeterminada en los archivos de pydroid. Después de descargarlo de la play store, volver a pydroid y hacer la instalación como se indica arriba en **Como instalar bibliotecas necesarias**. La liga a numpy para descargarla en la play store aparece cuando tratas de instalarlo desde el pip.



Introducción a Numpy

Numpy es una biblioteca de Python muy potente que nos ayuda a realizar operaciones con matrices de una forma sencilla y natural. Existen otras bibliotecas famosas, como TensorFlow de Google, pero suelen ser mas potentes, tardadas y especializadas para trabajos científicos dedicados.

<https://numpy.org>

Aplicación

La aplicación esta debidamente comentada en cada función para que se pueda comprender su funcionamiento.

Ocupa el funcionamiento de Gauss-Jordan visto en la página de Wikipedia acerca de este procedimiento (https://es.wikipedia.org/wiki/Eliminaci%C3%B3n_de_Gauss-Jordan).

El algoritmo es el siguiente:

1. Pide la cantidad de columnas y de renglones, viene especificado al correr el programa
2. Al generar la matriz y el vector a trabajar, pide los valores de estos, pueden ser numeros enteros o flotantes (con punto decimal), la biblioteca Numpy lo soporta
3. Procedimiento de eliminación escalonada
 - I. Es un proceso iterativo que empieza acomodando los renglones de modo que no quede 0 en el pivote. También pone como pivote al número con menor peso (para ello se compará con el valor absoluto) para hacer las operaciones de division y restas menos engorrosas. Tomando en cuenta que el pivote se da por Matriz[i,i] (por que el pivote debe ser un elemento de la diagonal) se compara a todos los renglones pero solo en el elemento i de cada uno para determinar cual es el mejor pivote de acuerdo a las reglas de arriba
 - II. Con el pivote encontrado, se normaliza el renglon i donde se encuentra el pivote, para ello solo se toma como factor de normalización al valor del pivote, se divide a todo el renglon entre este valor y con ello tendremos un pivote con valor igual a 1
 - III. Con el renglon del pivote normalizado, se multiplica este renglon por el valor i de cada renglon (el valor i de cada renglon vendría siendo el número de este renglon que se encuentra en la columna del pivote) y se resta el renglon del pivote a cada renglon para hacer ceros todos los valores i de los demás renglones
 - IV. Se repite este proceso pasando al siguiente valor de i, cambiando de columna para encontrar el pivote y pasando al siguiente renglon.
4. En caso de que un renglon se vuelva cero (indicando dependencia en el sistema de ecuaciones) el programa lo notificará eliminando los renglones con puros ceros.
5. En caso de que el número de ecuaciones sea menor al número de incognitas, el programa no arroja soluciones concretas. Por el contrario, si el numero de ecuaciones es igual al número de incognitas, el programa dará las soluciones que encontró. Si el número de ecuaciones es mayor al número de incognitas, puede pasar que una ecuación sea combinación lineal de las demás o que el sistema no tenga soluciones, en este último caso el programa deberá fallar.