Trabajo Práctico Nº 0

Números complejos, Linux y Julia

►**1)** Dado el siguiente número complejo en forma de fracción:

$$\frac{1}{1+i}\,.$$

a) Escriba la parte real y la parte imaginaria del complejo equivalente.

b) Calcule sus coordenadas polares (módulo y fase).

c) Grafique el círculo unidad en el plano complejo.

d) Grafique el vector que representa el complejo en el plano complejo.

e) ¿Está dentro o fuera del círculo unidad?

f) ¿En qué cuadrante se encuentra, I, II, III o IV?

g) Cambie le fase en 90° multiplicando por $e^{i\,\frac{\pi}{2}}$.

h) ¿En qué cuadrante se encuentra el nuevo valor?

i) ¿Por qué número complejo debe multiplicar para cambiar la fase en 180°?

2) Encuentre las 4 raíces complejas del siguiente polinomio complejo de grado 4:

$$P(Z) = Z^4 + 8\,Z^2 + 16\,.$$

3) Comandos básicos de Linux:

a) Abra una terminal en su `home` desde el menú de Aplicaciones del escritorio.

b) Cree un directorio en su home llamado `ASG`: `mkdir ASG`.

c) Descargue este Trabajo Práctico Nº 0 (`tp00.pdf`) desde la página web de la materia y guárdelo dentro de `ASG`.

d) Cambie de directorio hacia `ASG`: `cd ASG`.

e) Dentro de `ASG` cree un subdirectorio llamado `TP0`: `mkdir TP0`.

f) Retroceda un nivel de directorio mediante el comando: `cd ...`.

g) Mueva `tp00.pdf` dentro de `TP0` mediante: `mv tp00.pdf TP0/`.

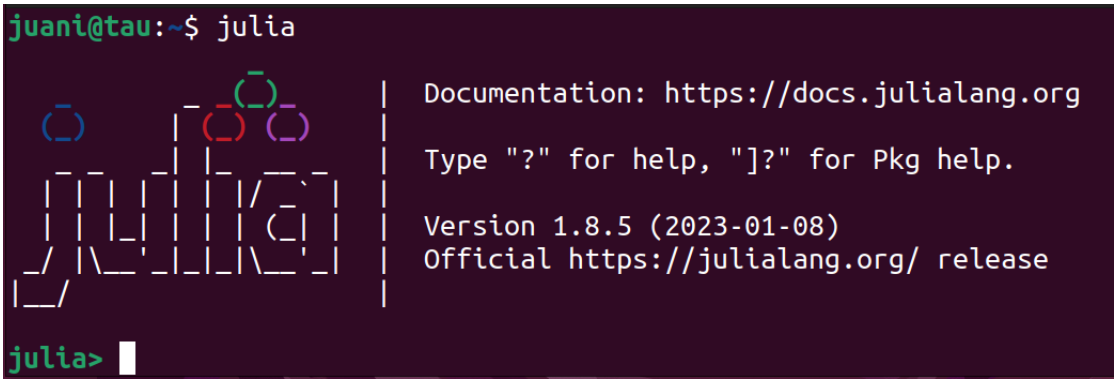
4) Consulte el material sobre el lenguaje `Julia` provisto en la página web de la materia:

a) Visite los tutoriales.

b) Opcional: lea los instructivos para instalar `Julia` en su computadora personal.

►5) Primeros pasos en `Julia`.

a) Abra `Julia` en una terminal tipeando en la terminar `julia` + Enter. Esto lo lleva al REPL (modo Read Eval Print Loop):



b) Entre al gestor de paquetes mediante el símbolo cerrar corchete `]`.

c) Instale el paquete `PolynomialRoots`: `add "PolynomialRoots"`.

d) Salga del gestor de paquetes con `backspace` (tecla borrar hacia atrás).

e) Llame al paquete "PolynomialRoots" para poder usarlo: `using PolynomialRoots`.

f) Defina la secuencia de coeficientes del polinomio $P(Z)$ del ejercicio 2, comenzando por el término independiente: `p = [16, 0, 8, 0, 1]`.
Observación: En `Julia` los vectores y demás arreglos se escriben definidos entre corchetes y no entre paréntesis como ocurre en otros lenguajes como Fortran, Octave o Python.

g) Calcule las raíces del polinomio $P(Z)$ mediante el comando `roots(p)`.

h) Calcule nuevamente las raíces de $P(Z)$, esta vez redondeando a 4 dígitos decimales utilizando la función `round()` así: `round.(roots(p), digits=4)`.
Observación: Note el punto "`.`" al final del comando `round`. La sintaxis de `Julia` impone usar un punto al final de cada comando que se aplica sobre arreglos y no sobre escalares.

i) ¿Coincide con el resultado que obtuvo en el ejercicio 2?

j) Salga de `Julia` con `exit()` o bien con `ctrl + d`.