1. Graficar sobre un mismo sistema de ejes (usando **matplotlib** en el dominio $[-1, 1]$) a la función $f(x) = e^x$ y a los polinomios de Taylor centrados en $a = 0$ de grados $1, 2, 3, 4$ y $5$. Personalice un título y distinga todas las curvas con estilo, color y etiqueta.

2. Graficar sobre un mismo sistema de ejes (usando **matplotlib** en el dominio $[-\pi, \pi]$) a la función $f(x) = \cos x$ y a los polinomios de Taylor centrados en $a = 0$ de grados $2, 4, 6$ y $8$. Personalice un título y distinga todas las curvas con estilo, color y etiqueta.

3. Graficar sobre un mismo sistema de ejes (usando **matplotlib** en el dominio $[-2\pi, 2\pi]$) a la función $f(x) = \sin x$ y a los polinomios de Taylor centrados en $a = 0$ de grados $1, 3, 5$ y $7$. Personalice un título y distinga todas las curvas con estilo, color y etiqueta.

4. Graficar sobre un mismo sistema de ejes (usando **matplotlib** en el dominio $[-\frac{1}{2}, \frac{1}{2}]$ ) a la función $f(x) = \frac{1}{x-1}$ y a los polinomios de Taylor centrados en $a = 0$ de grados $1, 2, 3, 4$ y $5$. Personalice un título y distinga todas las curvas con estilo, color y etiqueta.

5. Sea $f(x) = \cos(\pi x/4)$. Encuentre el polinomio $P$ de menor grado posible que toma los mismos valores que $f$ en los puntos $-2, -4/3, 0, 4/3, 2$. Luego grafique sobre un mismo sistema de ejes a $f$ y a $P$.

6. Sea $f(x) = \sin(\pi x/4)$. Encuentre el polinomio $P$ de menor grado posible que toma los mismos valores que $f$ en los puntos $-2, -4/3, 0, 4/3, 2$. Luego grafique sobre un mismo sistema de ejes a $f$ y a $P$.

7. Sea $f(x) = \cos(2\pi x)$. Encuentre el polinomio $P$ de menor grado posible que toma los mismos valores que $f$ en los puntos $-2, -4/3, 0, 4/3, 2$. Luego grafique sobre un mismo sistema de ejes a $f$ y a $P$.

8. Sea $f(x) = \sin(2\pi x)$. Encuentre el polinomio $P$ de menor grado posible que toma los mismos valores que $f$ en los puntos $-2, -4/3, 0, 4/3, 2$. Luego grafique sobre un mismo sistema de ejes a $f$ y a $P$.

9. Sea $f(x) = \dfrac{1}{1 + 4x^2}$ en el dominio $[-1, 1]$. Construya y grafique en un mismo sistema de ejes a $f$ y a los polinomios de interpolación $P$ y $Q$, usando para $P$ los puntos $[-0.8, -0.6, -0.4, -0.2, 0, 0.2, 0.4, 0.6, 0.8]$ y usando para $Q$ los ceros del polinomio de Chebyshev $T_9(x)$ en $[-1, 1]$. Personalice con título, estilo y etiquetas.

10. Sea $f(x) = \dfrac{1}{1 + 9x^2}$ en el dominio $[-1, 1]$. Construya y grafique en un mismo sistema de ejes a $f$ y a los polinomios de interpolación $P$ y $Q$, usando para $P$ los puntos $[-0.8, -0.6, -0.4, -0.2, 0, 0.2, 0.4, 0.6, 0.8]$ y usando para $Q$ los ceros del polinomio de Chebyshev $T_9(x)$ en $[-1, 1]$. Personalice con título, estilo y etiquetas.

11. Sea $f(x) = \dfrac{1}{1 + 16x^2}$ en el dominio $[-2, 2]$. Construya y grafique en un mismo sistema de ejes a $f$ y a los polinomios de interpolación $P$ y $Q$, usando para $P$ los puntos $[-0.16, -0.12, -0.8, -0.4, 0, 0.4, 0.8, 0.12, 0.16]$ y usando para $Q$ los ceros del polinomio de Chebyshev $T_9(x)$ en $[-2, 2]$. Personalice con título, estilo y etiquetas.

12. Sea $f(x) = \dfrac{1}{1 + 25x^2}$ en el dominio $[-3, 3]$. Construya y grafique en un mismo sistema de ejes a $f$ y a los polinomios de interpolación $P$ y $Q$, usando para $P$ los puntos $[-0.24, -0.18, -0.12, -0.6, 0, 0.6, 0.12, 0.18, 0.24]$ y usando para $Q$ los ceros del polinomio de Chebyshev $T_9(x)$ en $[-3, 3]$. Personalice con título, estilo y etiquetas.

13. Defina en **Python**, una función llamada **seminormaDeTaylor()**, que reciba como argumentos a expresión simbólica de sympy (representando a una función derivable $n$ veces en $a$), que reciba al flotante $a$, al entero $n$ y que tenga como salida el valor:

$$\|f\|_{Taylor} := \sum_{k=0}^{n} |f^{(k)}(a)|$$

14. Defina en **Python**, una función llamada **seminormaDeInterpolacion()**, que reciba como argumentos a expresión simbólica de sympy (representando a una función), que reciba una lista $\mathbf{X} = [x_0, x_1, \ldots, x_n]$ de flotantes distintos (representado a los puntos de interpolación) y que tenga como salida el valor:

$$\|f\|_{interpolación} := \sum_{k=0}^{n} |f(x_k)|$$

15. Defina en **Python**, una función llamada **normaCuadratica()**, que reciba como argumentos a expresión simbólica de sympy (representando a una función integrable en $[a, b]$ ), que reciba los flotantes $a$ y $b$ (representando los extremos del intervalo $[a, b]$ ) y que tenga como salida el valor:

$$\|f\|_{cuadrática} := \sqrt{\int_a^b |f(x)|^2 dx}$$

16. Defina en **Python**, una función llamada **normaMaximo()**, que reciba como argumentos a expresión simbólica de sympy (representando a una función continua en $[a, b]$ ), que reciba los flotantes $a$ y $b$ (representando los extremos del intervalo $[a, b]$ ) y que tenga como salida el valor:

$$\|f\|_{\infty} := \max_{a \leq x \leq b} |f(x)|$$

17. Use la función **seminormaDeTaylor()** para calcular la "distancia" entre:

    - $f(x) = e^x$ y $g(x) = 1 + x + \frac{1}{2}x^2$ con $a = 0$ y $n = 2$.
    - $f(x) = e^x$ y $g(x) = 1 + x + x^2$ con $a = 0$ y $n = 2$.
    - $f(x) = e^x$ y $g(x) = 1 + x + x^2$ con $a = \frac{1}{2}$ y $n = 2$.

    **Sug:** Recuerde que $d(f, g) = \|f - g\|$

18. Use la función **seminormaDeInterpolacion()** para calcular la "distancia" entre:

    - $f(x) = x^2 - 1$ y $g(x) = x(x - 1)(x + 1)$ con $X = [-1, 0, 1]$.
    - $f(x) = x(x^4 - 1)$ y $g(x) = x^3(x - 1)(x + 1)$ con $X = [-1, 0, 1]$.

    **Sug:** Recuerde que $d(f, g) = \|f - g\|$

19. Use la función **normaCuadratica()** para calcular la distancia entre:

    - $f(x) = x^2 - 1$ y $g(x) = x(x - 1)(x + 1)$ con $[a, b] = [-1, 1]$.
    - $f(x) = x(x^4 - 1)$ y $g(x) = x^3(x - 1)(x + 1)$ con $[a, b] = [-2, 2]$.

    **Sug:** Recuerde que $d(f, g) = \|f - g\|$

20. Use la función **normaMaximo()** para calcular la distancia entre:

    - $f(x) = x^2 - 1$ y $g(x) = x(x - 1)(x + 1)$ con $[a, b] = [-1, 1]$.
    - $f(x) = x(x^4 - 1)$ y $g(x) = x^3(x - 1)(x + 1)$ con $[a, b] = [-2, 2]$.

    **Sug:** Recuerde que $d(f, g) = \|f - g\|$