ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA EN SISTEMAS

METODOS NUMERICOS ICCD412

*JUAN FRANCISCO PINTO ANDRANGO*

*Tarea 6 Unidad 03-A Serie de Taylor y Polinomios de Lagrange*

*GR1CC*

*FECHA DE ENTREGA 30 DE NOVIEMBRE DEL 2025*

## Conjunto de Ejercicios

- deteermine el orden de la mejor aproximación para las siguientes funciones, usando la serie de Taylor Y el Polinomio de Lagrange:

1. $\frac{1}{25*x^2+1}$ ; $x_0 = 0$
2. $arctan(x), x_0 = 1$

- Escriba las Fórmu;as de los diferentes polinomios
- Grafique las diferentes Aproximaciones

## Series de Taylor

para $f(x) = \frac{1}{25x^2+1}$ $X_0 = 0$

$f'(x) = -\frac{50X}{(25x^2+1)^2}$

$f''(x) = -\frac{50(1-75X^2)}{(25x^2+1)^3}$

$f'''(x) = \frac{15000x(-25x^2+1)}{(25x^2+1)^4}$

$f''''(x) = \frac{15000(3125x^4-250x^2+1)}{(25x^2+1)^5}$

$f'''''(x) = \frac{3750000(-1875x^4+250x^2-3)}{(25x^2+1)^6}$

$f''''''(x) = \frac{3750000(328125x^6-65625x^4+1575x^2-3)}{(25x^2+1)^7}$

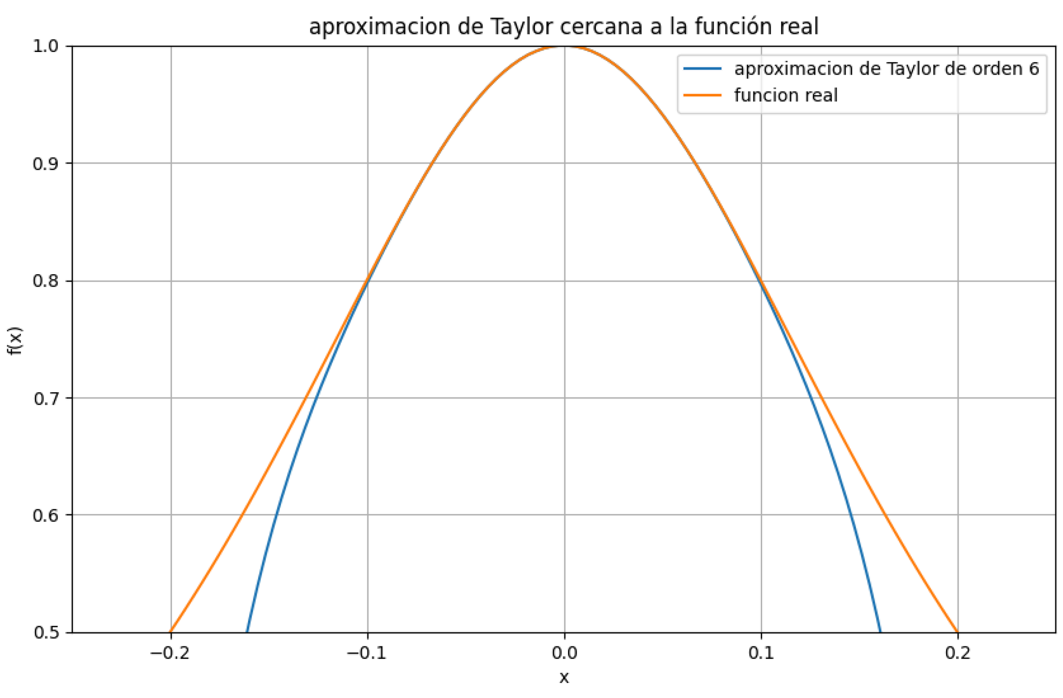
| i | fi (X_0) | P_i (x) |
|---|---|---|
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 2 | -50 | -25x^2 |
| 3 | 0 | 0 |
| 4 | 15000 | 625x^4 |
| 5 | 0 | 0 |
| 6 | -11250000 | -15625x^6 |

## Resultados

para obtener la arpoximación más cercana a la real se hizo $n = 6$ iteraciones , lo que nos dio como resultado un polinomio de 6to grado

$p(x) = 1 - 25x^2 + 625X^4 - 15625x^6$

In [6]:
```python
### grafico de las Series de Taylor

import matplotlib.pyplot as plt
import numpy as np
import sympy as sp

def f(x):
    return 1 / (25*x**2 + 1)

def f_taylor(x):
    return 1 - 25*x**2 + 625*x**4 - 15625*x**6

x = np.linspace(-0.2, 0.2, 100)

y = f_taylor(x)

plt.figure(figsize=(10, 6))
plt.title("aproximacion de Taylor cercana a la función real")
plt.plot(x, y, label="aproximacion de Taylor de orden 6")
plt.plot(x, f(x), label="funcion real")
plt.xlabel("x")
plt.ylabel("f(x)")
plt.grid()
plt.legend()
plt.xlim(-0.25, 0.25)
plt.ylim(0.5, 1)
plt.show()
```




## Series de Taylor

para $f(x) = arctan(x), con x_0 = 1$

$f'(x) = \frac{1}{(x^2+1)}$

$f''(x) = -\frac{2x}{(x^2+1)^2}$

$f'''(x) = -\frac{2-6x^2}{(x^2+1)^3}$

$f''''(x) = \frac{24(x-x^3)}{(x^2+1)^4}$

$f'''''(x) = \frac{24(-10x^2+5x^4+1)}{(x^2+1)^5}$

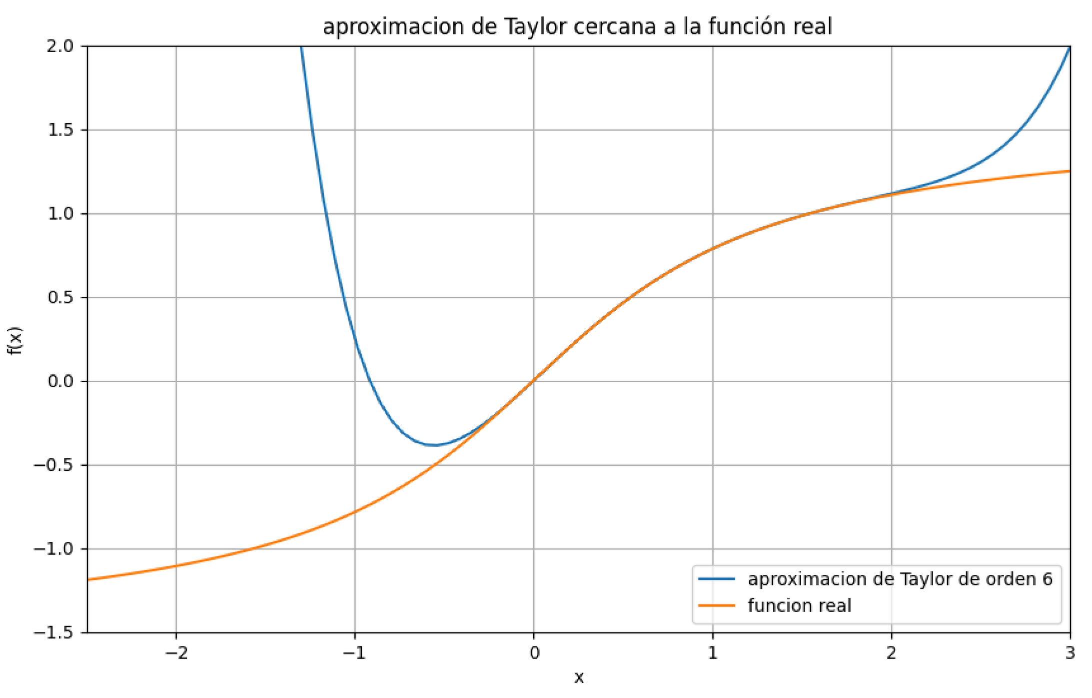
$f''''''(x) = \frac{24(100x^3-30x^5-30X)}{(x^2+1)^6}$

| i | $f^i(x_0)$ | $p_i(x)$ |
|---|---|---|
| 0 | $\frac{\pi}{4}$ | $\frac{\pi}{4}$ |
| 1 | 0.5 | $\frac{x-1}{2}$ |
| 2 | - 0.5 | $-\frac{(x-1)^2}{4}$ |
| 3 | 0.5 | $\frac{(x-1)^3}{12}$ |
| 4 | 0 | 0 |
| 5 | -3 | $-\frac{(x-1)^5}{40}$ |
| 6 | 15 | $\frac{(x-1)^6}{48}$ |

para obtener una aproximacion cercana se realiza 6 iteraciones y obtuvimos un polinomio de 6to grado

$P(x) = \frac{\pi}{4} + \frac{1}{2}(x-1) - \frac{1}{4}(x-1)^2 + \frac{1}{12}(x-1)^3 - \frac{1}{40}(x-1)^5 + \frac{1}{48}(x-1)^6$

In [8]:
```python
#### grafica de serie de taylor para la segunda función
def f_2(x):
    return np.arctan(x)

def f_2_taylor(x):
    return np.pi/4 + (x-1)/2 - (x-1)**2/4 + (x-1)**3/12 - (x-1)**5/40 + (x-1)**6/48

x = np.linspace(-np.pi, np.pi, 100)

y = f_2_taylor(x)
plt.figure(figsize=(10, 6))
plt.title("aproximacion de Taylor cercana a la función real")
plt.plot(x, y, label="aproximacion de Taylor de orden 6")
plt.plot(x, f_2(x), label="funcion real")
plt.xlabel("x")
plt.ylabel("f(x)")
plt.grid()
plt.legend()
plt.xlim(-2.5, 3)
plt.ylim(-1.5, 2)
plt.show()
```



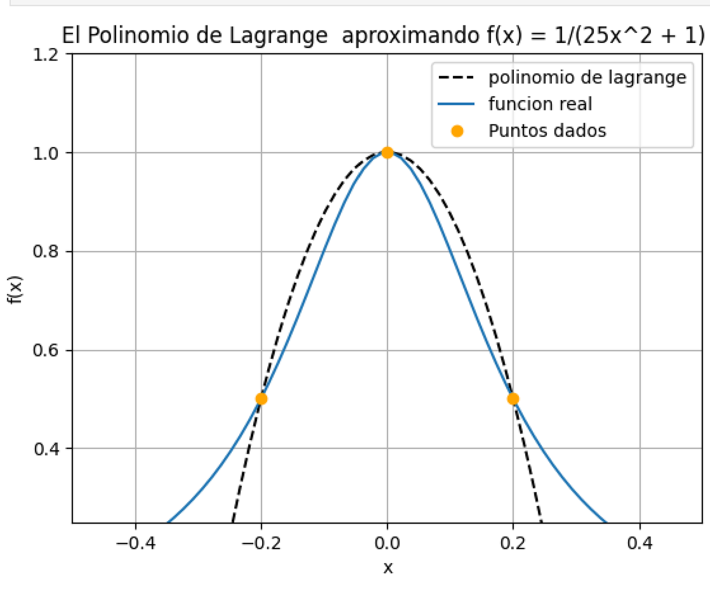

## Polinomios de Lagrange

$para f(x) = \frac{1}{25X^2+1}$

Usamos los puntos de la tabla para calcular el polinomio

| X | $-\frac{1}{5}$ | 0 | $-\frac{1}{5}$ |
|---|---|---|---|
| y | $-\frac{1}{2}$ | 1 | $\frac{1}{2}$ |

In [9]:
```python
#### primera funcion de polinomios de lagrange

#### para f(x)  = 1/(25x^2 + 1)

def polinomio(x):
    return (-0.08*x**2 + 0.0064) / 0.0064

points2 = [(-1/5, 1/2), (0, 1), (1/5, 1/2)]

x_lagrange = np.linspace(-3, 3, 400)
y_lagrange = polinomio(x_lagrange)

y_real = f(x_lagrange)

x_coords, y_coords = zip(*points2)
plt.plot(x_lagrange, y_lagrange, label='polinomio de lagrange', linestyle='--', color='black')
plt.plot(x_lagrange, y_real, label='funcion real')
plt.plot(x_coords, y_coords, 'o', label='Puntos dados', color='orange')
plt.legend()
plt.grid()
plt.title('El Polinomio de Lagrange  aproximando f(x) = 1/(25x^2 + 1) ')
plt.xlabel('x')
plt.ylabel('f(x)')
plt.xlim(-0.5, 0.5)
plt.ylim(0.25, 1.2)
plt.show()
```




## Polinomio de lagrande

$para f(x) = artan(x)$

usamos los puntos de la tabla para calcular el polinomio

| X | -2 | 0 | 1 | 2 |
|---|---|---|---|---|
| y | -1.107148718 | 0 | $\frac{\pi}{4}$ | 1.107148718 |

el polinomio de lagrande :

$P(x) = \frac{(-1.107148718)(x)(x^2-3x+2)}{-24} + \frac{(\pi)(x)(x^2-4)}{-12} + \frac{(1.107148718)(x)(x^2+x-2)}{8}$

In [ ]:
```python
def polinomio_2(x):
    return (-1.107148718)*x*(x**2-3*x+2) / (-24) + (np.pi*x*(x**2-4))/(-12) \
    + (1.107148718*x*(x**2+x-2)) / 8

points_2 = [(0, 0), (-2, -1.107148718), (1, np.pi/4), (2, 1.107148718)]
x_lagrange_2 = np.linspace(-3, 3, 100)
y_lagrange_2 = polinomio_2(x_lagrange_2)

x_coords_2, y_coords_2 = zip(*points_2)
plt.plot(x_lagrange_2, y_lagrange_2, label='polinomio de lagrange', color='black', linestyle='--')
plt.plot(x, f_2(x_lagrange_2), label='uncion real')
plt.plot(x_coords_2, y_coords_2, 'o', label='Puntos dados', color='orange')
plt.legend()
plt.grid()
plt.title('el polinomio de Lagrange para f(x) = arctan(x)')
plt.xlabel('x')
plt.ylabel('f(x)')
plt.xlim(-3, 3)
```

```
plt.ylim(-2, 2)
plt.show()
```