

Creado por: José L. Ayala

este prompt permite crear el programa de una clase habiéndole proporcionado la temática y la población objetivo de la misma.

Prompt:

Quiero que actúes como un profesor universitario de Ciencias Físicas. Tendrás que crear contenidos atractivos e informativos para materiales didácticos

Resultado:

Plan de Clase

¡Claro! Como profesor universitario de Ciencias Físicas especializado en Meteorología, es un placer ayudarte a

Creado por: José L. Ayala

Este prompt describe el logo generado para una empresa para la que hemos especificado su actividad, población objetivo, y características de la misma. Una versión reducida del texto es

Prompt:

escribe el logo de una empresa de cuidadores de perros a domicilio llamada "Wouaw". Los clientes de esta empresa son fundamentalmente hombres y mujeres de

Resultado:

El logo de "Wouaw", la empresa de cuidadores de perros a domicilio, debe transmitir una imagen sofisticada y confiable para atraer a los clientes que buscan un servicio exclusivo y de calidad

Generar código python para representar funciones con matplotlib

Ciencias

fecha de creación: Tue, 22 Aug 2023 18:10:21 GMT

Herramienta: Chat-GPT

Creado por: FRANCISCO JOSE GARATE SANTIAGO

Muchos conceptos matemáticos (como por ejemplo cálculo diferencial con varias variables, curvas de nivel u optimización con restricciones) pueden resultar difíciles de entender si no se

Prompt:

Quiero que actúes como un experto en Python.
¿Podrías generar el código para visualizar con la librería matplotlib la siguiente función paraboloidal $-(a * x^2 + b * y^2)$ y su plano perpendicular $-2 * a * x - 2 * b * y$ superpuesto en la misma gráfica? Usa colores naranjas.

Resultado:

```
# Definir la función del paraboloidal y el plano perpendicular
def paraboloidal(x, y, a, b):
    return -(a * x**2 + b * y**2)

def plano_perpendicular(x, y, a, b):
    return -2 * a * x - 2 * b * y
```

```
# Rango de valores para x e y
x = np.linspace(-5, 5, 100)
y = np.linspace(-5, 5, 100)
x, y = np.meshgrid(x, y)

# Valores de los parámetros 'a' y 'b' para la
función
a = 1
b = 1

# Calcular los valores del paraboloide y el
plano perpendicular
z_paraboloide = paraboloide(x, y, a, b)
z_plano = plano_perpendicular(x, y, a, b)

# Crear la figura y el eje 3D
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

# Graficar el paraboloide en naranja
ax.plot_surface(x, y, z_paraboloide,
cmap='Oranges', alpha=0.7)

# Graficar el plano perpendicular en naranja
con transparencia
ax.plot_surface(x, y, z_plano,
cmap='Oranges', alpha=0.5)

# Etiquetas de los ejes
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')

# Mostrar la gráfica
plt.show()
```

```
prompt_chatgptipyrb x +
[2]: import numpy as np
import matplotlib.pyplot as plt

# Definir los valores de 'a' y 'b' para el paraboloide y el plano
a = 1
b = 2

# Crear una malla de valores para 'x' y 'y'
x = np.linspace(-10, 10, 100)
y = np.linspace(-10, 10, 100)
x, y = np.meshgrid(x, y)

# Calcular los valores de 'z' para el paraboloide y el plano
z_paraboloide = -(a * x**2 + b * y**2)
z_plano = -2 * a * x - 2 * b * y

# Generar la gráfica
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

# Graficar el paraboloide en color naranja
ax.plot_surface(x, y, z_paraboloide, cmap='Oranges', alpha=0.7)

# Graficar el plano perpendicular en color naranja claro
ax.plot_surface(x, y, z_plano, cmap='Oranges', alpha=0.5)

# Mostrar la gráfica
plt.show()
```

