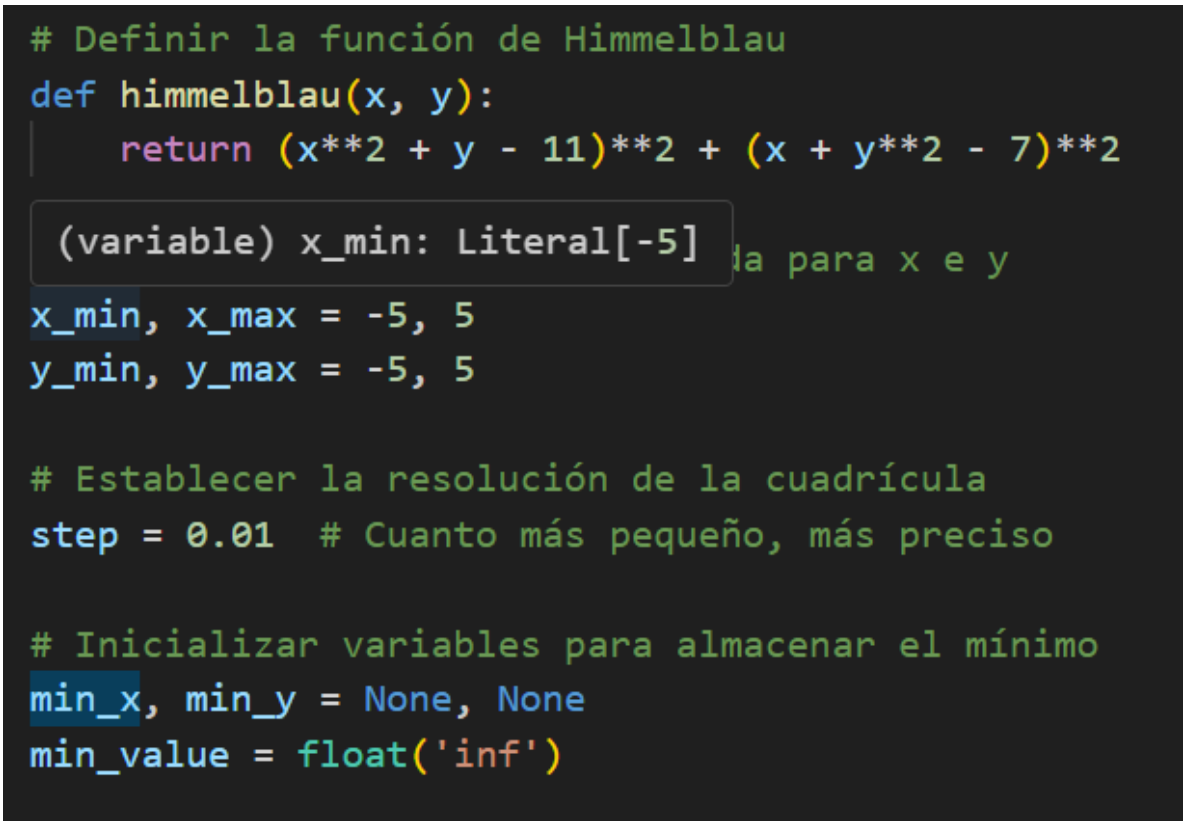
Juan carlos sandoval resendiz




```python
# Definir la función de Himmelblau
def himmelblau(x, y):
    return (x**2 + y - 11)**2 + (x + y**2 - 7)**2

x_min, x_max = -5, 5
y_min, y_max = -5, 5

# Establecer la resolución de la cuadrícula
step = 0.01  # Cuanto más pequeño, más preciso

# Inicializar variables para almacenar el mínimo
min_x, min_y = None, None
min_value = float('inf')
```

Declare una función que resolvera la función de Himmelblau que se le pasaran los valores x, y

Se declara el rango de y_min , y_max y x_max, x_min desde -5 hasta 5

Se declara un paso que se usara próximamente para encontrar valores mas precisamente

Se declara min_x y min_y que almacenara los valores mínimos para resolver la función y un min_value que guardara el resultado de la función

Juan carlos sandoval resendiz

```python
# Búsqueda por fuerza bruta en la cuadrícula
x = x_min
while x <= x_max:
    y = y_min
    while y <= y_max:
        value = himmelblau(x, y)
        if value < min_value:
            min_value = value
            min_x, min_y = x, y
        y += step
    x += step

# Mostrar los resultados
print(f"Los valores de (x, y) que minimizan la función son: ({min_x}, {min_y})")
print(f"Valor mínimo de la función: {min_value}")
```

Se resuelve por fuerza bruta en donde entramos a un bucle donde manejaremos como una cuadricula que se recorrerá paso por paso y evaluando la función para encontrar el mejor valor

X se le asigna el valor x_min y entramos al while si x es menor o igual a x_max se asigna un valor de y como y_min y entramos al while si y es menor o igual a y_max

Se evalua la función con los valores contenidos y si el valor obtenido es menor al min_value se almacenan los valores se avanza un paso y hasta que se iguale a y_max y ahora avanzamos un paso en x y regresamos al bucle de y para revisar los valores