

```

import psycopg2
from osgeo import gdal
import io

def get_image_extent(image_path):
    dataset = gdal.Open(image_path)
    if not dataset:
        raise ValueError("Não foi possível abrir a imagem.")

    geotransform = dataset.GetGeoTransform()
    width = dataset.RasterXSize
    height = dataset.RasterYSize

    # Calcula os limites espaciais da imagem
    minx = geotransform[0]
    maxx = geotransform[0] + width * geotransform[1]
    miny = geotransform[3] + height * geotransform[5]
    maxy = geotransform[3]

    # Retorna como um WKT POLYGON
    extent = f"POLYGON(({minx} {miny}, {maxx} {miny}, {maxx} {maxy}, {minx} {maxy}, {minx} {miny}))"
    return extent

def store_satellite_image(db_config, image_path, image_name):
    try:
        # Lê a imagem como bytes
        with open(image_path, 'rb') as f:
            image_data = f.read()

        # Obtém o extent da imagem
        extent_wkt = get_image_extent(image_path)

        # Conecta ao banco de dados
        conn = psycopg2.connect(
            host = db_config['host'],
            dbname = db_config['dbname'],
            user = db_config['user'],
            password = db_config['password'],
            port = db_config.get('port', 5432) # Porta padrão do PostgreSQL
        )
        cursor = conn.cursor()

        # Inserção da imagem e do extent
        query = """
        INSERT INTO satellite_images (name, image, extent)
        VALUES (%s, %s, ST_GeomFromText(%s, 4326))
        """
        cursor.execute(query, (image_name, psycopg2.Binary(image_data), extent_wkt))
        conn.commit()

        print(f"Imagem '{image_name}' armazenada com sucesso no banco de dados!")

    except Exception as e:
        print(f"Erro ao armazenar a imagem: {e}")

    finally:
        if cursor:
            cursor.close()
        if conn:
            conn.close()

# Configuração do banco
db_config = {
    'host': 'localhost',
    'dbname': 'banco_postgis',
    'user': 'usuario',
    'password': 'senha',
    'port': 5432
}

```

```
# Caminho da imagem
image_path = '/GIS/imagens/imagem_GeoTIFF.tif'
image_name = 'Imagem de Satélite GeoTIFF'

# Armazena a imagem
store_satellite_image(db_config, image_path, image_name)
```