

# Parte 1

```
In [ ]: import os
import numpy as np
from PIL import Image
import torchvision
import torchvision.transforms as transforms
from torch.utils.data import DataLoader, Dataset
import torch

# Configuración de CUDA si está disponible
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

# Ruta a la carpeta que contiene Las imágenes originales
data_folder = 'img_align_celeba/img_align_celeba'

# Número de imágenes que deseas utilizar (por ejemplo, 10,000)
num_images = 200000

# Carpeta donde guardarás Las imágenes preprocesadas
output_folder = 'images'

# Crear la carpeta de salida si no existe
if not os.path.exists(output_folder):
    os.makedirs(output_folder)

# Transformaciones para el preprocesamiento de imágenes
transform = transforms.Compose([
    transforms.Resize((128, 128)),
    transforms.ToTensor(),
    transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5)) # Normalización
])

class CelebADataset(Dataset):
    def __init__(self, root_dir, transform=None, num_images=None):
        self.root_dir = root_dir
        self.transform = transform
        self.image_paths = os.listdir(root_dir)[:num_images]

    def __len__(self):
        return len(self.image_paths)

    def __getitem__(self, idx):
        img_name = os.path.join(self.root_dir, self.image_paths[idx])
        image = Image.open(img_name)
        if self.transform:
            image = self.transform(image)
        return image

# Crear una instancia del conjunto de datos
celeba_dataset = CelebADataset(data_folder, transform=transform, num_images=num_images)

# Crear un DataLoader para cargar Las imágenes en Lotes
```

```
batch_size = 64
data_loader = DataLoader(celeba_dataset, batch_size=batch_size, shuffle=True, num_w

# Preprocesar y guardar todas las imágenes en la carpeta de salida
image_counter = 0 # Inicializar un contador para nombres únicos de imágenes
for batch in data_loader:
    for i in range(batch.size(0)):
        image = batch[i]
        save_path = os.path.join(output_folder, f'image_{image_counter}.jpg')
        torchvision.utils.save_image(image, save_path)
        image_counter += 1 # Incrementar el contador para el siguiente nombre de i

print(f'Se han preprocesado y guardado {image_counter} imágenes en {output_folder}.
```

Se han preprocesado y guardado 200000 imágenes en images.