

In [ ]:

```
import numpy as np                # per manipolare array e matrici
from PIL import Image            # per manipolare le immagini
from matplotlib import pyplot as plt # per creare e visualizzare grafici

def forward_diff_process(img_t_meno_1, beta, t):
    """ Tale funzione implementa la generica transizione del processo
    di diffusione in avanti dall'immagine al passo t-1 all'immagine
    al passo t il processo di diffusione in avanti.

    Input:
        img_t_meno_1: Image at the previous timestep (t - 1)
        beta: Scheduled Variance
        t: Passo temporale corrente
    Output:
        Immagine ottenuta al passo temporale corrente
    """

    beta_t = beta[t].reshape(-1, 1, 1)

    # 2. Calcolo media statistica e deviazione standard
    mu = np.sqrt((1.0 - beta_t)) * img_t_meno_1
    sigma = np.sqrt(beta_t)

    # 3. Generazione immagine al passo t tramite l'equazione (15)
    img_t = mu + sigma * np.random.randn(*img_t_meno_1.shape)
    #print(img_t_meno_1.shape)
    return img_t

# Esempio di applicazione del processo di diffusione in avanti

# 1. Si carica l'immagine di partenza (x0) usando la libreria PIL
img = Image.open("../content/prova.jpg")

# 2. Si ridimensiona l'immagine secondo le dimensioni desiderate
IMG_SIZE = (128, 128)
img = img.resize(size=IMG_SIZE)

# 3. Si definisce il numero dei passi temporali (T)
timesteps = 100

# 4. Generate beta (variance schedule)
beta_iniziale = 0.0001
beta_finale = 0.05
beta = np.linspace(beta_iniziale, beta_finale, num=timesteps, dtype=np.float32)

immagini_processate = []
img_t = np.asarray(img.copy(), dtype=np.float32) / 255.

# 5. Run the forward process to obtain img after t timesteps
for t in range(timesteps):
    img_t = forward_diff_process(img_t_meno_1=img_t, beta=beta, t=t)
    if t%20==0 or t==timesteps - 1:
        immagine = (img_t.clip(0, 1) * 255.0).astype(np.uint8)
        immagini_processate.append(immagine)

# 6. Si visualizzano le immagini fv
_, ax = plt.subplots(1, len(immagini_processate), figsize=(15, 6))

for i, immagine in enumerate(immagini_processate):
    ax[i].imshow(immagine)
    ax[i].set_title(f"Timestep: {i*20}")
```

```
ax[i].axis("off")
ax[i].grid(False)

plt.suptitle("Processo di diffusione in avanti", y=0.75)
plt.show()
plt.close()
```