

---

## **Clase 159 — Modelos de difusión (+ Stable Diffusion XL, ControlNet, LCM)**

Parte: 2 — Deep Learning · Fuente: Géron, cap. 17 § Diffusion Models + Ho et al. (2020) DDPM + papers SDXL, ControlNet, LCM. Duración estimada: 105 min.

# Clase 159 — Modelos de difusión (+ Stable Diffusion XL, ControlNet, LCM)

Parte: 2 — Deep Learning · Fuente: Géron, cap. 17 § Diffusion Models + Ho et al. (2020) DDPM + papers SDXL, ControlNet, LCM. Duración estimada: 105 min.

## Objetivo

Entender modelos de difusión — la familia que destronó a GANs en 2022 (DALL-E 2, Stable Diffusion, Midjourney). Idea: forward process agrega ruido gaussiano gradualmente; reverse process (aprendido) lo elimina paso a paso. Conocer DDPM clásico, latent diffusion (Stable Diffusion), ControlNet para condicionamiento espacial, LCM (Latent Consistency Models) para inference rápida.

## Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Implementar un DDPM sencillo: forward  $q(x_t | x_0)$ , U-Net que predice el ruido  $\epsilon_\theta(x_t, t)$ .
- Aplicar el sampling DDPM: 1000 steps de denoising desde  $x_T \sim N(0, I)$ .
- Reconocer latent diffusion: aplicar difusión en el espacio comprimido de un VAE (1/64 del tamaño) → 8× más rápido.
- Usar Stable Diffusion XL con diffusers library: prompt → imagen en 3 líneas.
- Aplicar ControlNet para condicionar generación en pose, edge, depth, segmentation.
- Acelerar inference con LCM o Turbo (1-4 steps en lugar de 50).

## Temas

- Forward process:  $q(x_t | x_{t-1}) = N(\sqrt{1-\beta_t} x_{t-1}, \beta_t I)$ .
- Closed form:  $q(x_t | x_0) = N(\sqrt{\alpha_t} x_0, (1-\alpha_t) I)$ .
- Loss simple (Ho 2020):  $MSE(\epsilon, \epsilon_\theta(x_t, t))$  — predecir el ruido.
- U-Net architecture (encoder-decoder con skip connections).
- Sampling DDPM vs DDIM vs DPM-Solver: 1000 → 50 → 20 steps.
- Complemento moderno: Stable Diffusion XL, ControlNet, LCM/Turbo.

## Versión profundizada — 2026

El tema moderno que antes vivía como complemento dentro de esta clase ahora tiene su(s) clase(s) propia(s) con patrón completo, ejercicios y homework:

- Clase 133a — Stable Diffusion XL + ControlNet en profundidad

## Definiciones y características

- Forward (diffusion) process:  $q(x_t | x_{t-1})$  agrega ruido. Markov chain.
- Reverse process:  $p_\theta(x_{t-1} | x_t)$  aprendido, denoise.
- Schedule:  $\beta_t$  o  $\alpha_t$  — cuánto ruido en cada step. Linear o cosine.

- U-Net: arquitectura encoder-decoder con skip connections, usada como denoiser.
- DDIM: sampler deterministic más rápido que DDPM (Song et al. 2021).
- Classifier-Free Guidance (CFG): en inference, mezcla condicional + incondicional para control.
- ControlNet: adapter para control espacial.
- LCM: distillation a 1-4 steps.

## Dataset / recursos

- Fashion-MNIST como playground.
- HF Hub para modelos pre-trained.
- Librerías: diffusers, transformers, accelerate, controlnet\_aux.

## Ejercicios

1. DDPM en MNIST: U-Net chico + DDPM scheduler. Entrenar 50 épocas; samplear 64 imágenes.
2. SDXL inference: cargar SDXL y generar 4 imágenes desde prompts.
3. CFG: variar guidance\_scale {1, 5, 15}. Ver cómo afecta fidelidad al prompt vs diversidad.
4. ControlNet Canny: tomar una foto, extraer bordes con OpenCV, generar variantes condicionadas con SDXL+ControlNet.
5. LCM: comparar SDXL base (30 steps) vs SDXL + LCM-LoRA (4 steps). Tiempo y calidad.

## Homework verificable

Pipeline de generación SDXL controlado:

1. Imagen base (e.g., una foto de un edificio).
2. Extraer Canny con OpenCV.
3. Generar 3 variantes con SDXL+ControlNet con prompts distintos.
4. Comparar tiempo SDXL vs SDXL+LCM.

Criterio de aceptación: las 3 variantes preservan la estructura del edificio según los Canny; LCM es  $\geq 5\times$  más rápido con calidad razonable.

## Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
OOM al cargar SDXL en GPU 8 GB	Modelo muy grande. Fix: fp16, enable_atten
Imágenes borrosas con LCM	Steps insuficientes. Fix: subir a 4-8; aju
ControlNet no obedece el control	controlnet_conditioning_scale muy bajo. Fi
Generación con prompts pobres da resultado	Prompt engineering es real. Fix: estilo "a
Difusión propia entrenada genera ruido	No converge. Fix: verificar la schedule, U

## Preguntas frecuentes

¿GAN o Diffusion en 2026?

Diffusion gana en calidad y controlabilidad. GAN (StyleGAN3) sigue siendo competitivo para caras (1 forward pass).

¿DDPM o DDIM o DPM-Solver?

Para inference: DPM-Solver / DPM++ (Lu et al. 2022) — 20 steps con misma calidad que DDPM 1000. Default en diffusers.

¿Cómo fine-tunear SDXL en estilo propio?

LoRA sobre el U-Net con diffusers examples. 20-50 imágenes alcanza para un estilo. ~1 hora en GPU consumer.

¿Difusión para video?

Sí. Sora, Veo, Stable Video Diffusion. Más caro computacionalmente.

¿Difusión para texto?

Sí pero menos competitivo que LLMs autoregresivos. Áreas activas de investigación.

## Referencias

- Géron, cap. 17 — Diffusion Models.
- Ho, Jain & Abbeel (2020), Denoising Diffusion Probabilistic Models, NeurIPS.
- Rombach et al. (2022), Stable Diffusion, CVPR.
- Zhang, Rao & Agrawala (2023), ControlNet, ICCV.
- Luo et al. (2023), Latent Consistency Models.
- diffusers docs.

## Siguiente clase

Clase 160 — Stable Diffusion XL + ControlNet en profundidad

## Apéndice: notebook (primer bloque)

Primera celda ejecutable del notebook de la clase.

```
# Imports y configuración inicial
```

## Archivos complementarios

- notebook.ipynb