**Hugo Córdoba Leal**

# LAB 5. **Ejemplos y casos de uso reales sobre la puesta en producción de soluciones de IA.**

**1. A la hora de solventar la restricción del uso de datos para desarrollar los modelos de IA, se apostó por la implementación de un módulo de sintetización de datos. ¿Qué alternativas propondrías a esta sintetización de datos?**

Además de la sintetización de datos, existen otras alternativas viables. Transfer Learning es una técnica que utiliza modelos preentrenados en grandes conjuntos de datos y ajusta (fine-tuning) esos modelos con los datos específicos del proyecto, reduciendo la necesidad de grandes volúmenes de datos. Data Augmentation implica generar más ejemplos a partir de los datos existentes mediante técnicas como la rotación, recorte, variación de brillo y contraste en imágenes, o la adición de ruido en datos de texto. Federated Learning permite entrenar modelos en dispositivos locales, compartiendo solo los parámetros del modelo, lo que protege la privacidad de los datos. Estas alternativas pueden ser más adecuadas dependiendo del contexto específico del problema y las restricciones de datos.

**2. ¿Qué opciones a nivel de arquitectura de modelo hubieras elegido actualmente para el desarrollo de la solución?**

Para el desarrollo de soluciones de IA, varias arquitecturas modernas pueden ser consideradas. Los modelos basados en transformers como BERT, GPT-3 o Vision Transformers (ViTs) son ideales para tareas de procesamiento de lenguaje natural y visión por computadora, captando relaciones complejas en los datos. Las Redes Neuronales Convolucionales (CNNs), como EfficientNet o ResNet, son robustas para problemas de visión por computadora y ofrecen eficiencia mejorada. Redes Neuronales Recurrentes (RNNs) y LSTM, aunque menos comunes ahora frente a transformers, son útiles para datos secuenciales y series temporales. Los autoencodersson útiles para la reducción de dimensionalidad y la representación compacta de datos. La elección de la arquitectura depende del tipo de datos y la tarea específica, buscando siempre un balance entre complejidad y rendimiento.

**3. ¿De qué manera hubieras solventado los límites de velocidad de ejecución en la inferencia de la solución?**

Optimizar la velocidad de inferencia es crucial para aplicaciones en tiempo real. Quantization convierte modelos a versiones que utilizan menos bits por parámetro, como INT8 en lugar de FLOAT32, reduciendo el tiempo de inferencia sin una perdida significativa de precisión. Pruning elimina conexiones y neuronas innecesarias en la red, haciendo el modelo mas ligero y rápido. La optimización de hardware utilizando mejores GPUs, TPUs o chips específicos mejoraría el rendimiento. Los modelos híbridos combinan inferencias rápidas en el borde con modelos mas complejos en la nube para garantizar mayor precisión.