

## Pre-requisitos

1. **Nota:** este taller se debe realizar en **grupos de tres personas**.
2. **Nota 2:** la entrega de este taller consiste en un **cuaderno de jupyter** ejecutable en Google Colab. El cuaderno debe contener todo el código y respuestas a las preguntas propuestas.

## Parte 1: Regresión con redes neuronales

Tome como base el modelo de redes neuronales presentado en el cuaderno de la clase para predecir el consumo de gasolina (MPG) a partir de los datos `auto-mpg.data` que encontrará en Bloque Neón.

1. Cree un nuevo modelo usando PyTorch que emplee como entrada una sola de las variables predictoras. Ajuste los datos de entrada y la red neuronal apropiadamente. El modelo debe contener una sola capa con neuronas lineales.
2. **Entrene** este modelo (éste será nuestro modelo base), **grafique** el historial de pérdida de entrenamiento, use los datos de prueba para **evaluar** el modelo calculando el MSE y el MAE, y genere una gráfica que muestre el modelo entrenado como una función lineal así como las predicciones para los datos de prueba (dispersión). Comente sus resultados.
3. Modifique el modelo para incluir una **función de activación no lineal**, entrene el modelo, genere los mismos resultados de evaluación y compare. Use siempre los mismos datos de entrenamiento y prueba. Comente sus resultados.
4. Modifique el **número de capas**, entrene el modelo resultante y compare. Use siempre los mismos datos de entrenamiento y prueba. Comente sus resultados.

## Parte 2: Clasificación con redes neuronales

En Bloque Neón encontrará los datos `heart.csv`, cuya descripción puede encontrar en <https://archive.ics.uci.edu/dataset/45/heart+disease>.

1. Cree un modelo de clasificación de redes neuronales usando PyTorch que emplee estos datos para predecir la ocurrencia de enfermedad cardíaca.
2. Entrene el modelo, grafique el historial de pérdida de entrenamiento, y evalúe el modelo usando métricas apropiadas para problemas de clasificación sobre los datos de prueba. Comente sus resultados.
3. Modifique el número de neuronas (2 casos diferentes al modelo base), entrene los modelos resultantes y compare. Use siempre los mismos datos de entrenamiento y prueba. Comente sus resultados.

4. Modifique el número de capas (2 casos diferentes al modelo base), entrene los modelos resultantes y compare. Use siempre los mismos datos de entrenamiento y prueba. Comente sus resultados.
5. Modifique las funciones de activación (2 casos diferentes al modelo base), entrene los modelos resultantes y compare. Use siempre los mismos datos de entrenamiento y prueba. Comente sus resultados.