

# Inteligencia Artificial en .NET

## Integrantes

Castillo Emiliano

Chilon Eduar

Fernández Santiago

Giménez Agustina

Ramos Juan Manuel

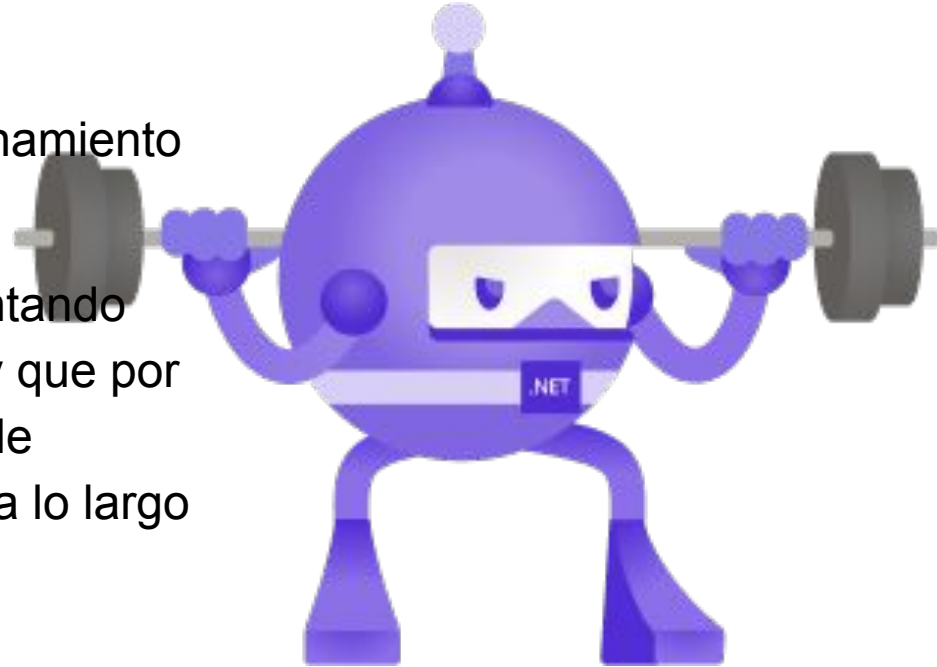


**Grupo 7**

# NUESTROS OBJETIVOS

Familiarizarnos con conceptos y el funcionamiento acerca de la inteligencia artificial.

Desarrollar una aplicación **.NET** implementando inteligencia artificial a través de **ML.NET** y que por la cual se pueda realizar reconocimiento de imágenes, persistir sus datos y utilizarlos a lo largo del proyecto.



# ¿Qué es la IA?

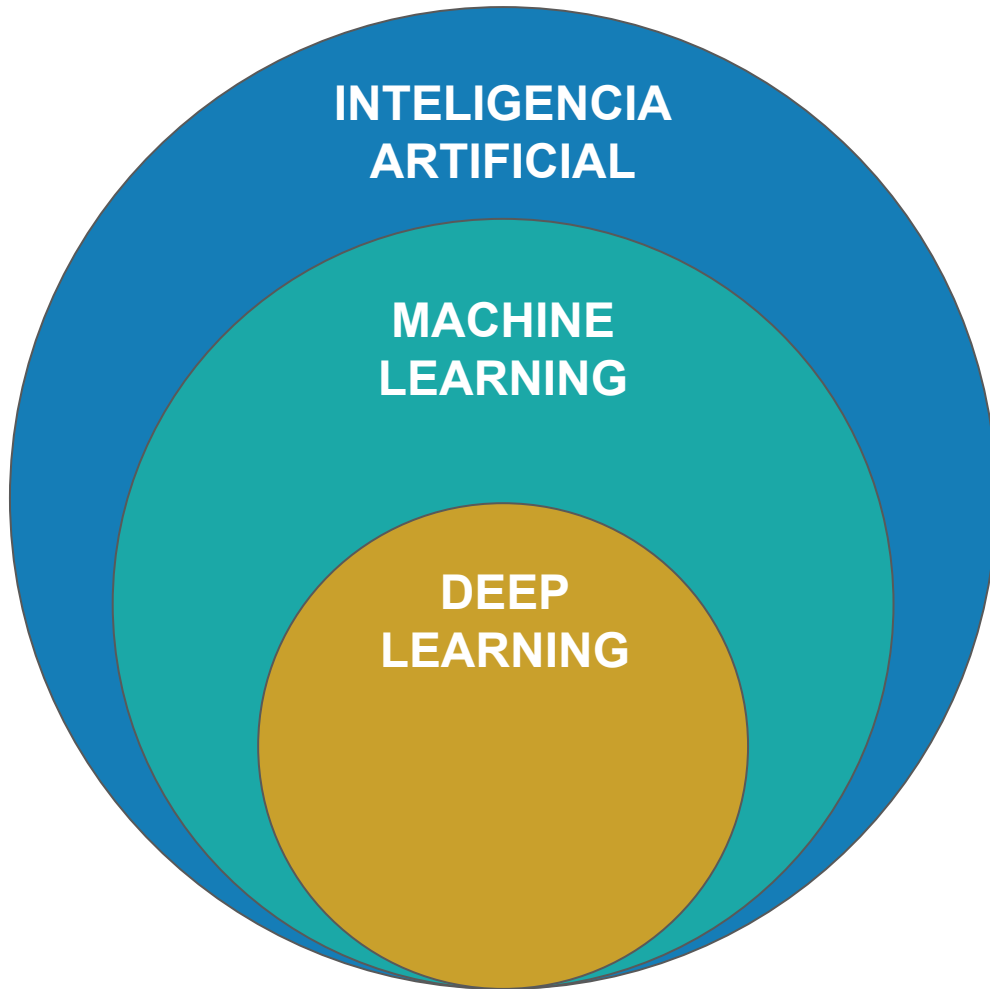
Podemos dividirla en los siguientes conceptos:

**Inteligencia Artificial.**

**Machine Learning.**

**Deep Learning.**





La inteligencia artificial son programas con la habilidad para aprender y razonar como humanos.

Machine Learning o Aprendizaje Automático son algoritmos con la habilidad de aprender sin ser programados explícitamente.

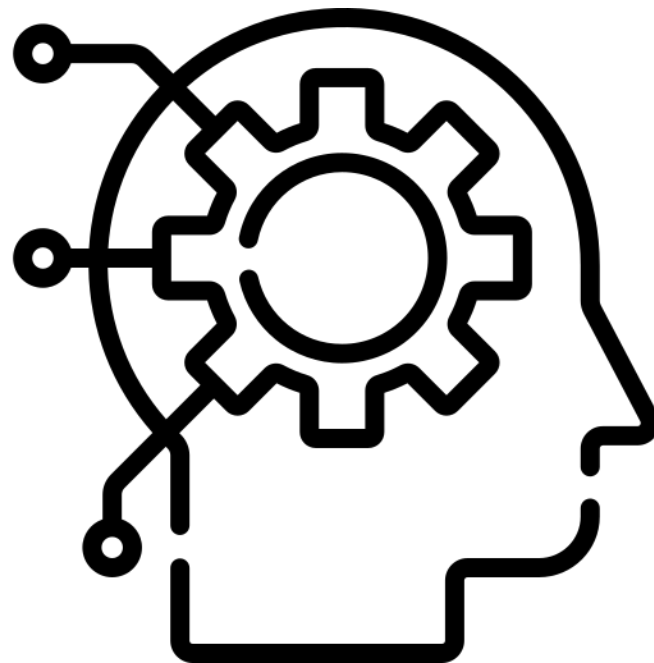
El Deep Learning es un subconjunto de aprendizaje automático en el que las redes neuronales artificiales se adaptan y aprenden de grandes cantidades de datos.

# MACHINE LEARNING

Es una rama de la inteligencia artificial que se enfoca en el desarrollo de algoritmos y modelos que permiten a las computadoras aprender y mejorar su rendimiento en tareas específicas a través de la experiencia. Estos algoritmos se diseñan para analizar datos, identificar patrones y tomar decisiones o hacer predicciones sin ser programados explícitamente para realizar una tarea en particular.

Es una disciplina más amplia que abarca diferentes enfoques algorítmicos para enseñar a las máquinas a aprender.

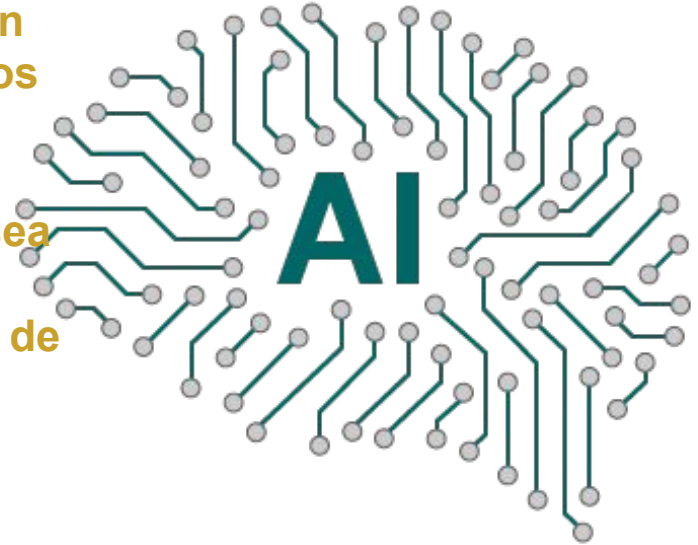
Un ejemplo se encuentra en el mundo del marketing, es la personalización de sugerencias en base a tu histórico, como hace Netflix o Amazon.



# DEEP LEARNING

Es una subárea del Machine Learning que se basa en redes neuronales artificiales con múltiples capas. Los modelos son capaces de aprender características y representaciones directamente de los datos sin una selección manual. Esto hace que el Deep Learning sea especialmente efectivo en tareas que involucran grandes cantidades de datos, como reconocimiento de imágenes, procesamiento de voz y traducción automática.

Deep Learning es una técnica específica dentro del Machine Learning que utiliza redes neuronales profundas para extraer y aprender representaciones de datos complejos.



# ¿Qué es ML.NET?

ML.NET es un framework de aprendizaje automático desarrollado por Microsoft que permite a los desarrolladores de software integrar fácilmente capacidades de aprendizaje automático en sus aplicaciones. Utiliza C# como lenguaje de programación.

ML.NET ofrece soporte para la implementación de modelos entrenados tanto localmente como en la nube, lo que brinda flexibilidad a los desarrolladores para adaptarse a sus necesidades. Además, se integra bien con otras bibliotecas y herramientas populares de aprendizaje automático, como TensorFlow y ONNX, lo que permite a los desarrolladores aprovechar modelos pre-entrenados y obtener mejores resultados.



# Nuestro proyecto e investigación.

Desarrollamos a partir de ML.NET una aplicación capaz de reconocer estados de ánimo a través del reconocimiento facial de imágenes, ya sea cargando imágenes locales o capturarlas en tiempo real.

## Model Builder

ML.NET proporciona una herramienta de interfaz gráfica llamada ModelBuilder para simplificar el proceso de construcción de modelos de aprendizaje automático. Está diseñado para desarrolladores con poca o ninguna experiencia en aprendizaje automático, lo que facilita la creación de modelos para escenarios comunes. Con Model Builder, puedes experimentar rápidamente con diferentes algoritmos, transformaciones de datos y arquitecturas de modelos para encontrar el mejor enfoque para tu problema.



# Y ¿Cómo funciona?

Los pasos para el funcionamiento de ML.NET se dividen en:

- Preparación de Datos.
- Cargar y pre-procesar el conjunto de datos.
- Diseño y flujo de trabajo del modelo.
- Entrenamiento del modelo.
- Evaluación del modelo.
- Implementación y uso.



## Preparación de datos:

En esta etapa, se recopilan y preparan los datos necesarios para entrenar el modelo. ML.NET proporciona herramientas para cargar y transformar los datos en un formato adecuado para el entrenamiento del modelo.

## Diseño y entrenamiento del modelo:

En esta etapa, se selecciona un algoritmo de aprendizaje automático adecuado para el problema que se está abordando. ML.NET ofrece una amplia gama de algoritmos para diferentes tipos de tareas, como clasificación, regresión y agrupación. Los desarrolladores definen la estructura del modelo y lo entrenan utilizando los datos preparados en la etapa anterior.

## Evaluación del modelo:

Después de entrenar el modelo, se evalúa su rendimiento utilizando datos de prueba. Esto implica medir métricas como la precisión, el error o el puntaje F1 para determinar qué tan bien se desempeña el modelo en datos no vistos. ML.NET proporciona herramientas para realizar esta evaluación y optimizar el modelo si es necesario.

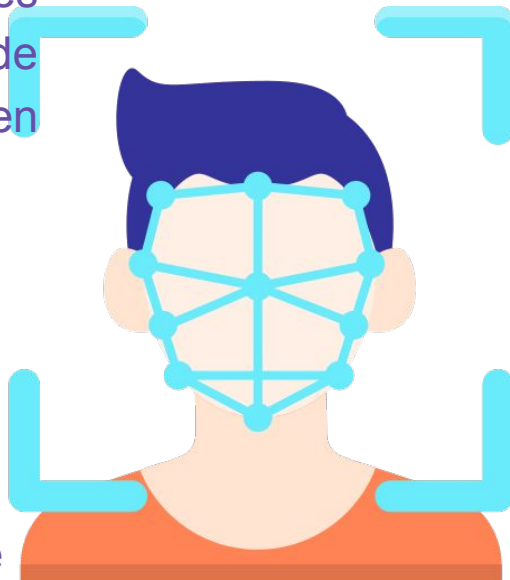
## Implementación y uso:

Una vez que el modelo ha sido entrenado y evaluado, se puede implementar en una aplicación .NET. ML.NET facilita la integración del modelo en la lógica de la aplicación, lo que permite realizar predicciones en tiempo real o utilizar el modelo para otros propósitos específicos.

# Memoria Descriptiva:

Comenzamos generando una clasificación de posibles emociones (feliz, enojado, triste, etc.) recopilando un conjunto de imágenes relacionadas con cada emoción y las organizamos en carpetas separadas.

Paso siguiente, entrenamos la inteligencia. Así, ML .NET puede detectar a través de patrones que existen en esas carpetas, cómo deberá clasificar imágenes que obtenga en un futuro. Una vez que entrenamos el modelo, automáticamente se genera un código para ser utilizado como api y así poder consumirlo desde nuestro proyecto.

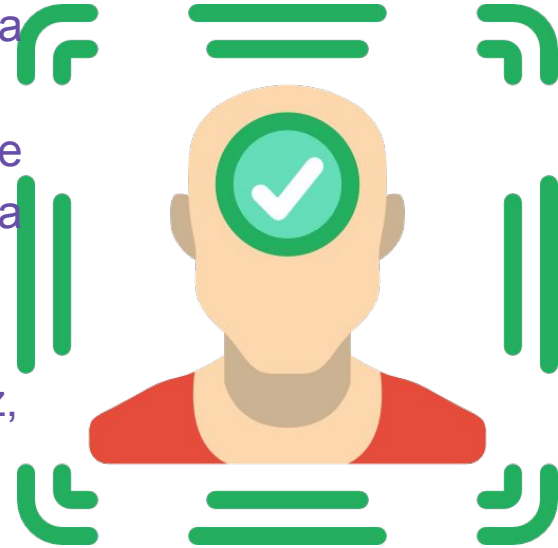


Una vez clasificadas las emociones según imágenes que fueron definidas para el entrenamiento del Modelo, procedimos a incorporar la cámara web (a través de JavaScript) en una nueva vista.

Una vez que se enciende la cámara, con un botón “Capturar” se toma una foto en momento real, la cual es clasificada por la inteligencia de igual manera que si fuera una imagen importada.

Todas las imágenes y sus respectivas clasificaciones (feliz, enojado, etc) son persistidas en una base de datos SQL Server.

Por último, creamos una vista de Resultados donde se obtendrá los datos persistidos en la base de datos.



# Conclusión:

Durante el desarrollo de este proyecto, nos propusimos familiarizarnos con los conceptos de Machine Learning y reconocimiento de imágenes utilizando la librería ML.Net. Nuestro objetivo principal fue desarrollar una aplicación que pudiera clasificar imágenes en base a emociones específicas, como felicidad, enojo y tristeza.

Aprendimos a utilizar ML.NET para entrenar un modelo, y pudimos aplicarlos con éxito en la clasificación de imágenes según diferentes emociones. Este proyecto nos brindó una comprensión más profunda de los conceptos de Machine Learning y su aplicación en el reconocimiento de imágenes, así como la capacidad de utilizar ML.Net de manera efectiva en futuros proyectos relacionados con este ámbito.

**Muchas gracias.**