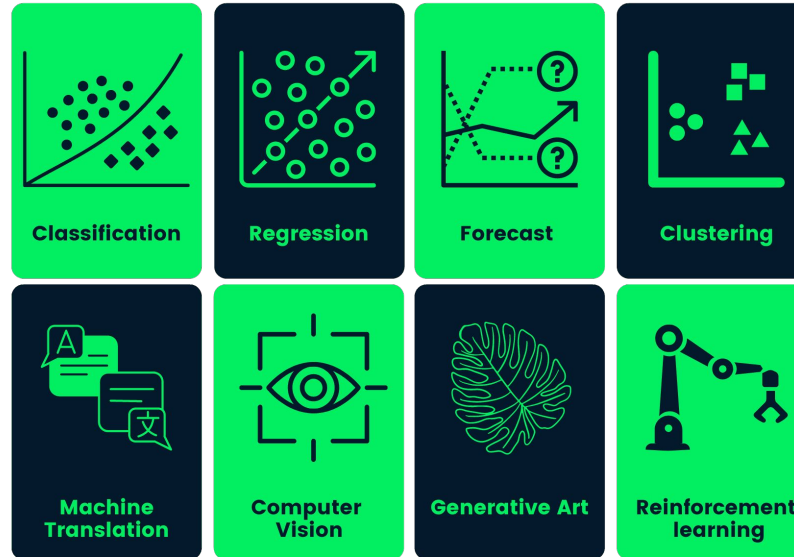


Introducción a la Inteligencia Artificial



Contenido resumido de la asignatura

Primer parcial:

- *Introducción a la Inteligencia Artificial y Agentes*
- *Lógica y Razonamiento Automático*
- *Resolución de Problemas mediante búsquedas*

Segundo parcial:

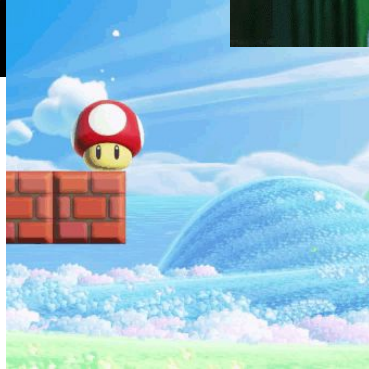
- *Lógica Difusa*
- *Modelos de Machine Learning*

Tercer parcial:

- *Deep Learning*
- *GEN AI*



Pero... ¿Te gustaría aprender con ellos?



Y no te preocupes, también aprenderás...

A *analizar sentimientos* en comentarios de productos y servicios

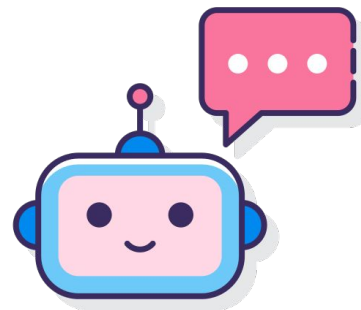


Identificar el género de un usuario por su *huella de voz*

Predecir si una empresa es susceptible a quebrar



Clasificar imágenes para sistemas de navegación autónoma



A hacer tu propio *ChatBot* para atención a clientes

¿Qué usaremos durante este curso?

Herramientas:

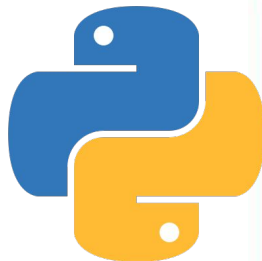
- Uso *OBLIGATORIO* de computadora en *CADA* sesión

IDEs (Local y OnLine):

- Visual Studio Code
- Pycharm
- Google Colab

Ambientes:

- Python
- Tensorflow
- Sklearn
- Numpy, pandas, Etc.



Horario de las sesiones

Día	Horario	Aula / formato
Lunes	7:00 PM - 8:30 PM	Remoto
Miércoles	7:00 PM - 8:30 PM	Salón A2

En todas las sesiones se pasará asistencia, y el alumno debe cumplir con la asistencia mínima para tener derecho a examen

Evaluación (Propuesta)

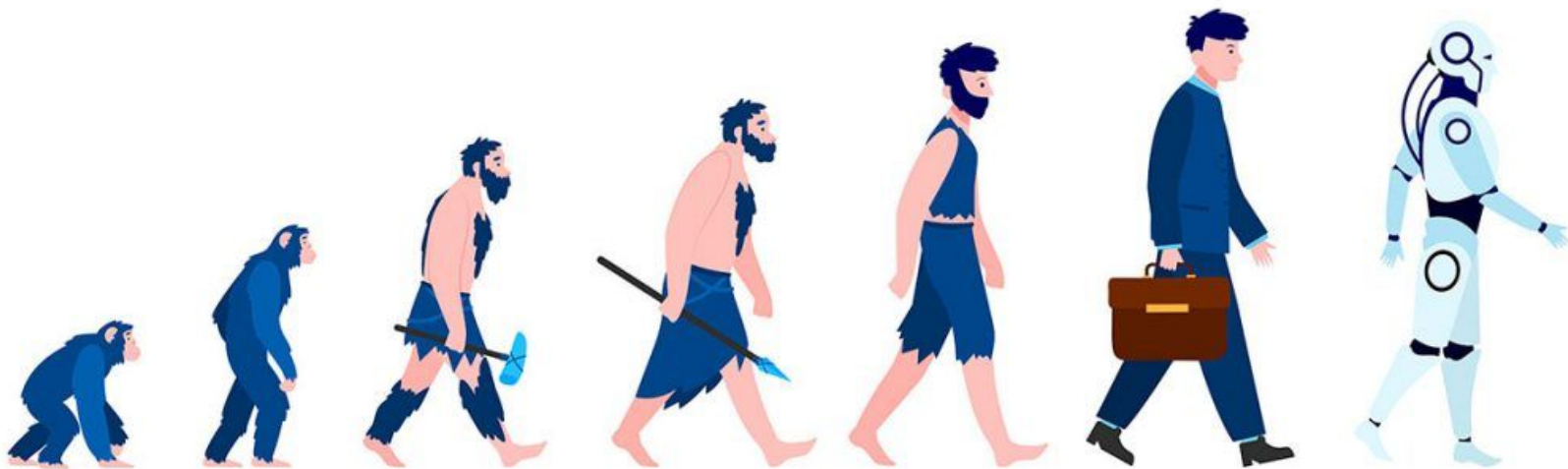
Parcial	Examen teórico	Examen práctico	Tareas / Prácticas	Proyecto	Total
1ro	10%	10%	10%	-	30%
2do	10%	10%	10%	-	30%
3ro	20%	-	10%	20%	50%

El curso se basa más en la práctica que en la teoría, y es posible obtener hasta 11 de calificación.

Los exámenes teóricos son 100% presenciales, y los prácticos se revisarán en el salón de clases.

El proyecto final se realizará en equipos.

Historia de la Inteligencia Artificial



Línea del tiempo de la Inteligencia Artificial

364 - 322 a.C.

1936

1956

1959

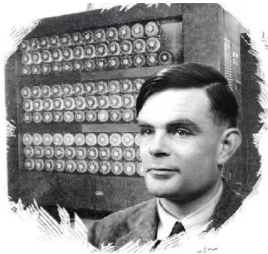
1965

2006

2014

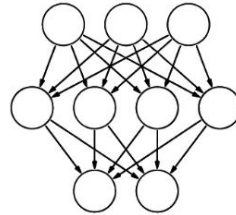
Aristóteles propone
los primeros
teoremas de lógica

Alan Turing propone una
máquina que pueda aprender



Se *inventa* el término
"Inteligencia Artificial"
(Dartmouth EU)

Se crea la *primer red*
neural "Adaline"



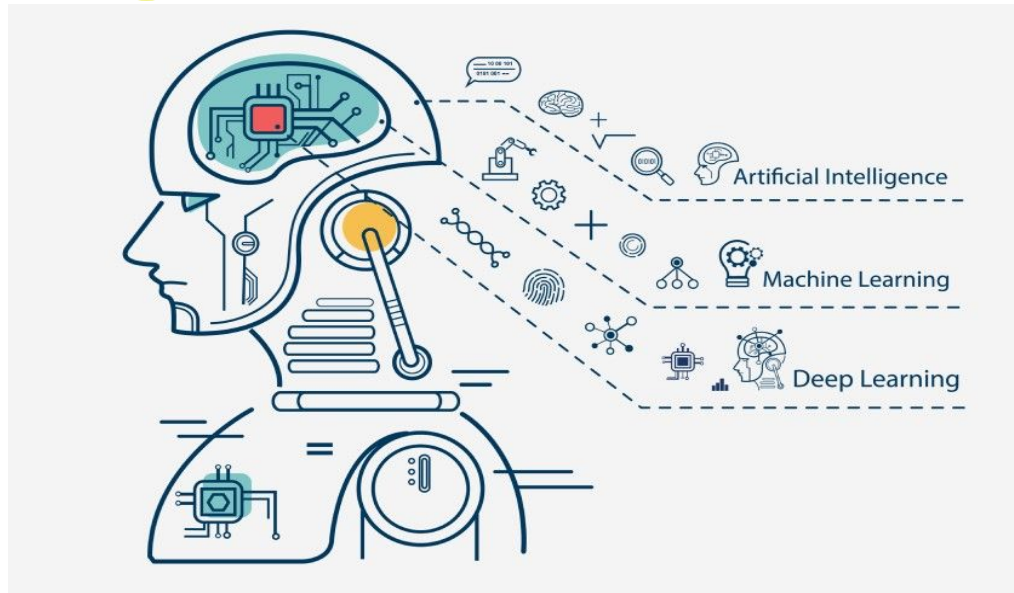
Se inventan los *buscadores*
heurísticos y los *conjuntos*
difusos

Joffrey Hinton inventa el
término "Deep Learning"



Se crean las *redes*
generativas GAN

Definición de Inteligencia Artificial



¿Qué es la Inteligencia Artificial?

Es cualquier **técnica, algoritmo, o método** que permita **tomar decisiones y crear** a partir de **probabilidades**. La IA intenta **simular** el proceso de decisiones **humano**, basado en **lógica, hechos, y ejemplos**. La IA puede:

Predecir / Clasificar

<https://teachablemachine.withgoogle.com/train/image>

Entender

<https://chat.openai.com/>

Generar

<https://this-person-does-not-exist.com/en>

Los niveles de la Inteligencia Artificial

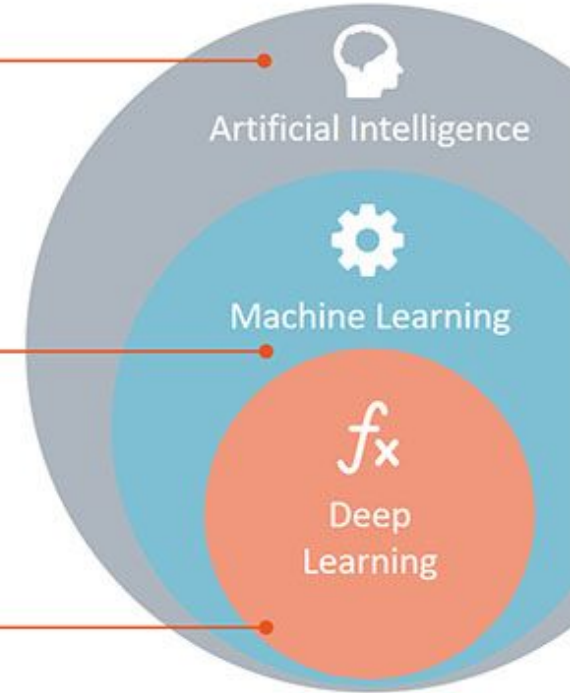
Algoritmos que **imitan el razonamiento humano**

(Sistemas basados en reglas, Sistemas expertos, Lógica proposicional, Lógica difusa, Métodos de búsqueda)

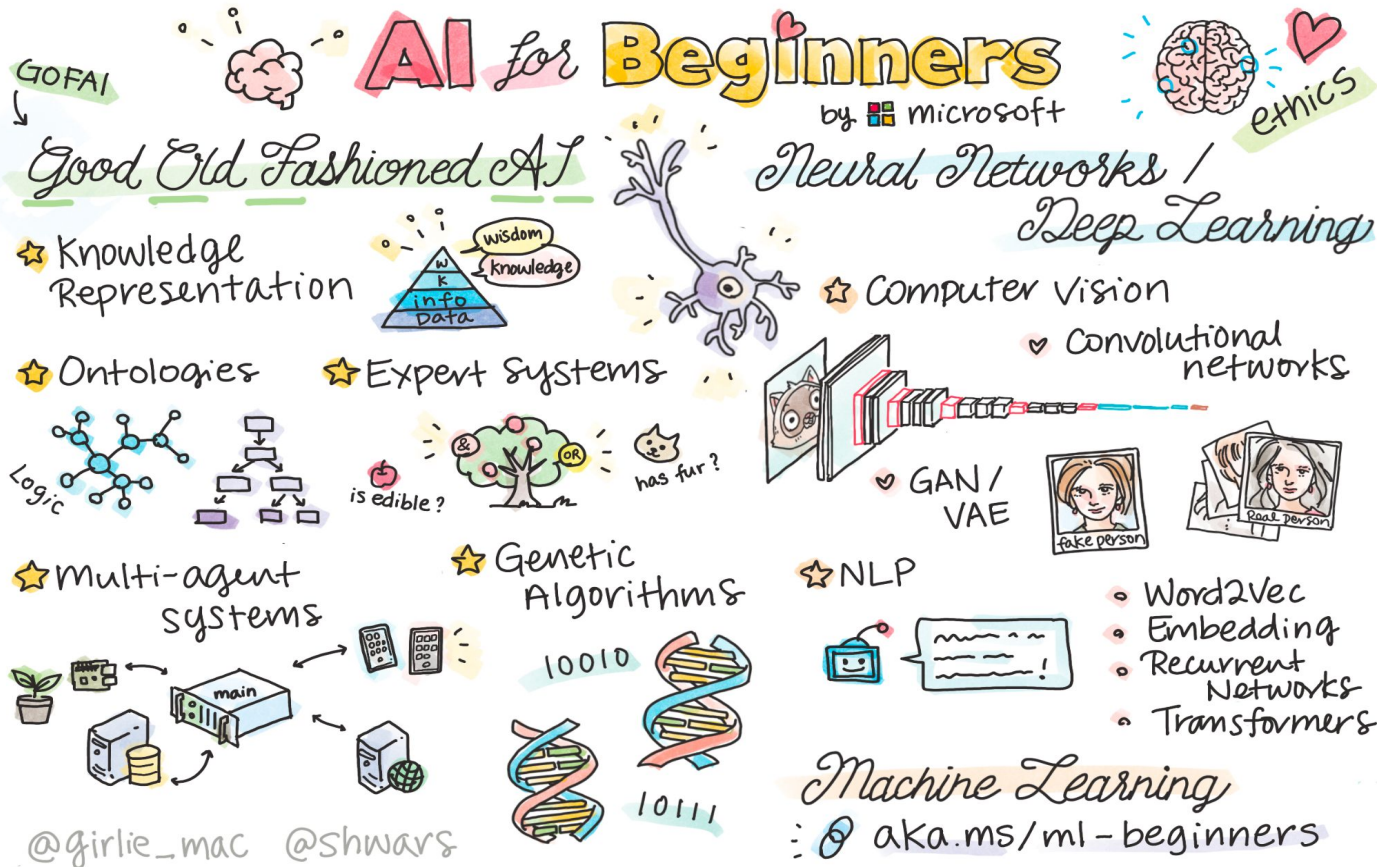
Algoritmos estadísticos que **aprenden de ejemplos**

(Máquinas de soporte vectorial, Regresiones lineales, Regresión logística, Árboles de decisión)

Redes neuronales multicapa que analizan profundamente (DNN, CNN, RNN, Transformers)

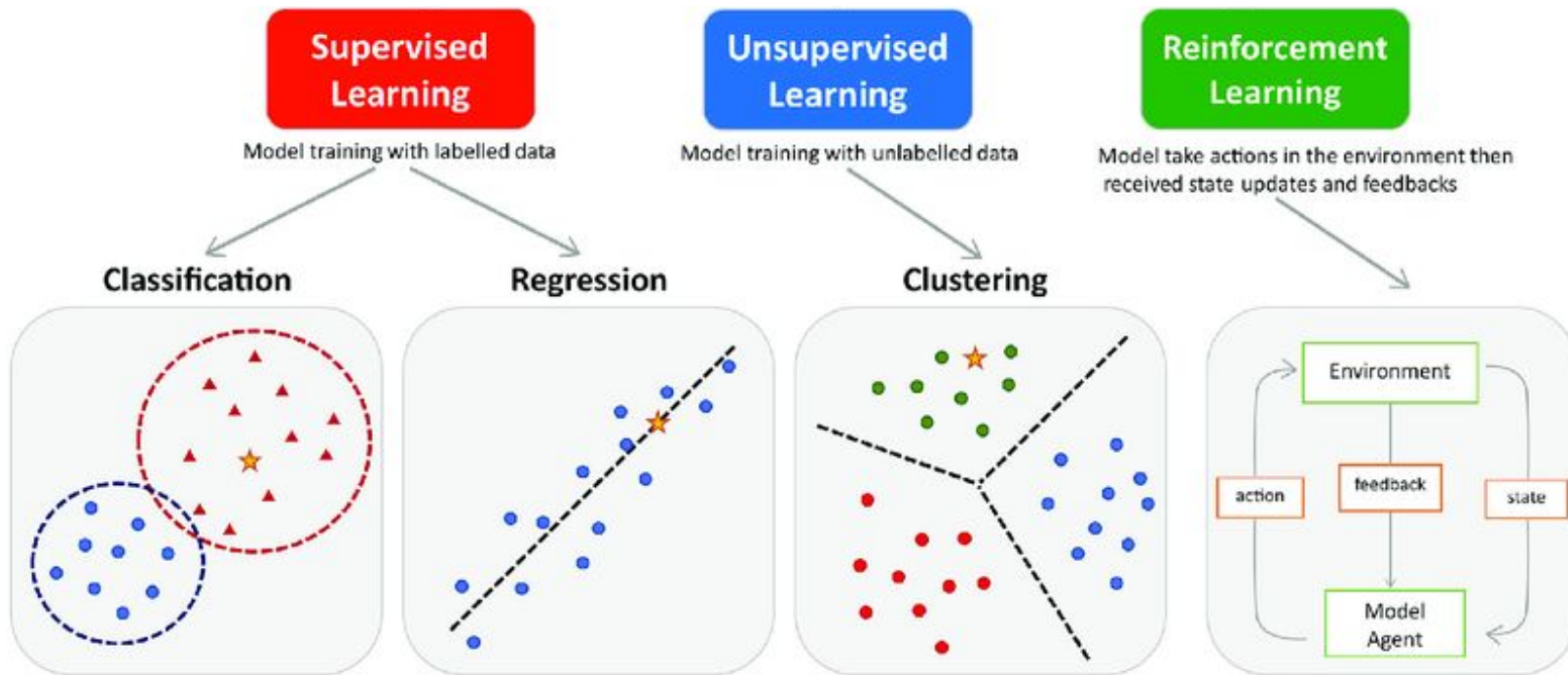


Algoritmos de Inteligencia Artificial

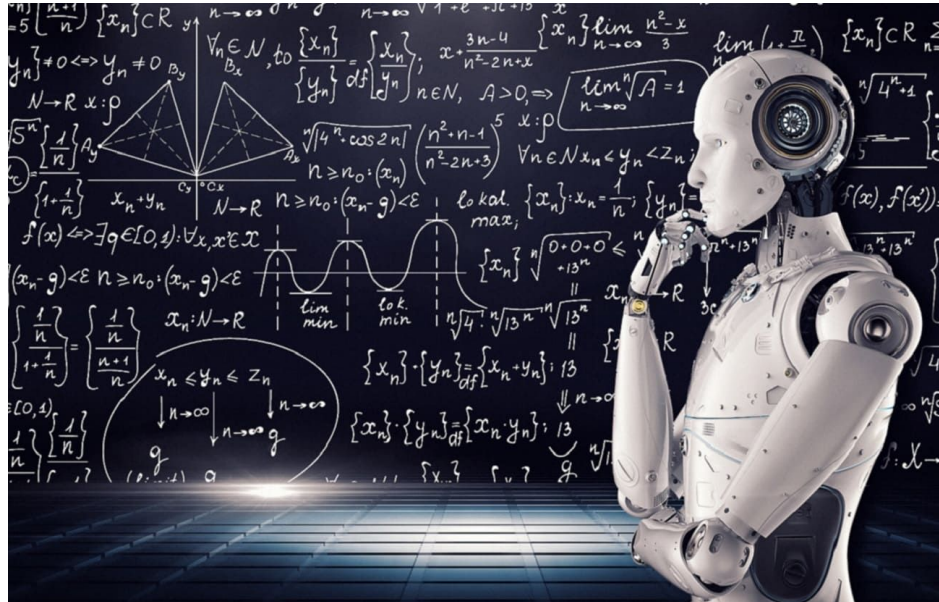


Tipos de Machine Learning

Los algoritmos de Machine Learning se dividen *principalmente en 3 tipos* según la *naturaleza del problema* que desean atacar:



Las 4 aproximaciones de la IA



Aproximaciones de la IA

La IA tiene diferentes **aproximaciones** según el **enfoque** que se le quiera dar, y se puede dividir en 4 sectores:

Humano

Racionalmente

Actuar

Actuar como humano.
Ejemplo: ChatGPT



Actuar racionalmente.
Ejemplo: Autos Autónomos

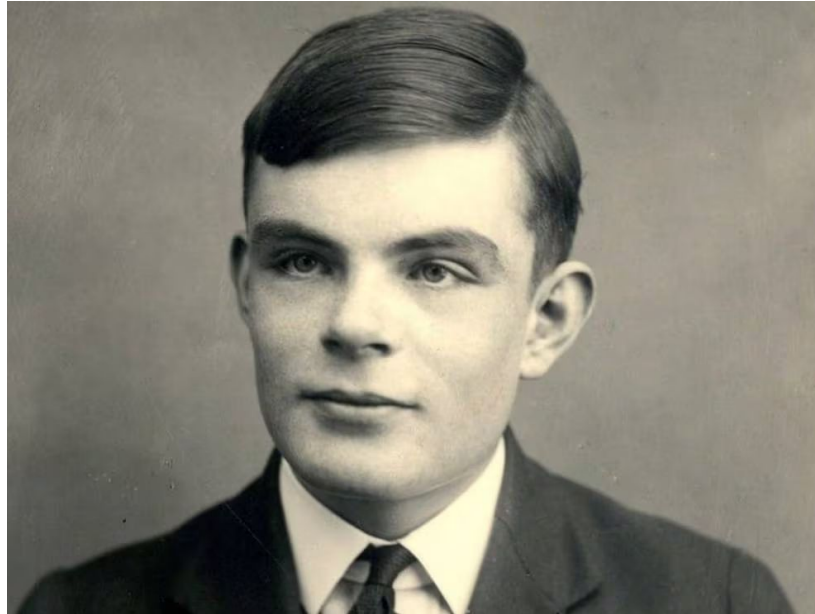
Pensar

Pensar como humano.
Ejemplo: Análisis de
sentimientos (Clasificación)

Pensar racionalmente o
idealmente.
Ejemplo: Sistemas expertos
(Deep Blue)

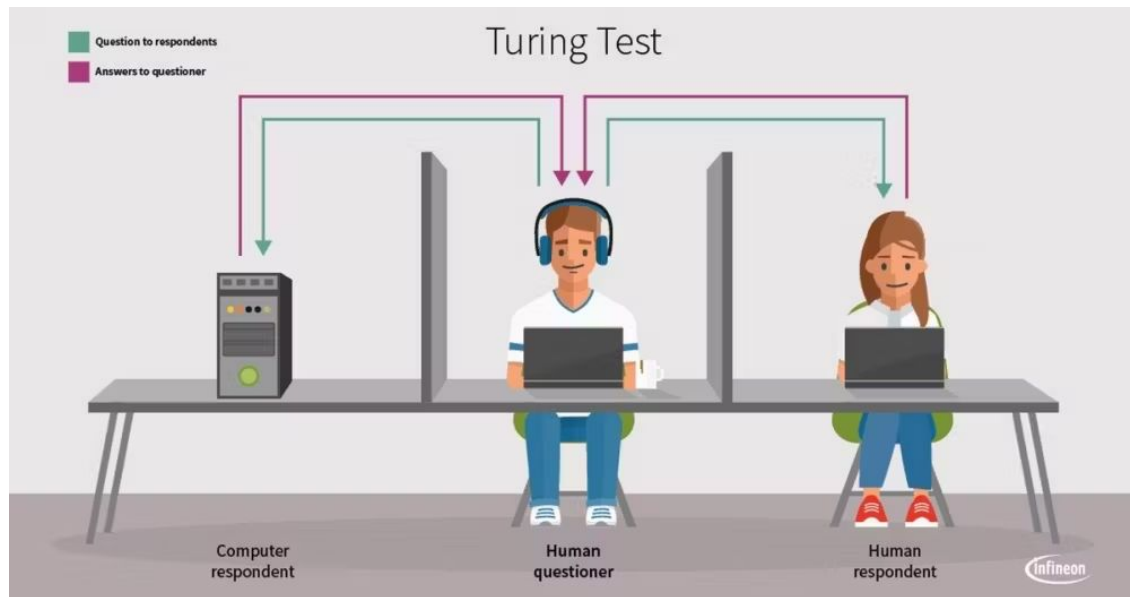
El Test de Turing

(*Actuar como humano*)



¿Qué es el test de Turing?

Si estuvieras platicando con un Robot, ¿Sabrías que estás platicando con un robot?
¿Cómo podrías distinguir una máquina de un humano?



El Test de turing consiste en interactuar en un entorno donde No sabemos si el interlocutor es un humano o una máquina, y así, medir qué tan posible es que el humano crea que está hablando con un humano cuando no es así

<https://www.humanornot.ai/>

<https://detective.stefanbohacek.dev/>

<https://chat.openai.com/>

Actuar como Humano

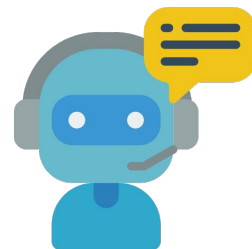
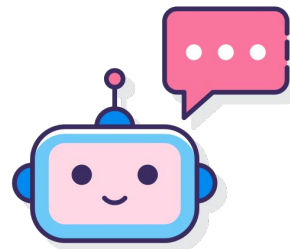
Los modelos bajo el enfoque de “actuar como humano”, generalmente son **modelos conversacionales** que pretenden contestar lo más **similar a un humano** posible. El mayor exponente actual es **ChatGPT**

“Hey hola, tengo una duda,
¿Cómo puedo comprar un
boleto para Canadá?”

No te preocupes, yo te explico,
puedes hacerlo de varias
formas, si quieres hacerlo por
internet, entonces primero ve a
la página... y luego...

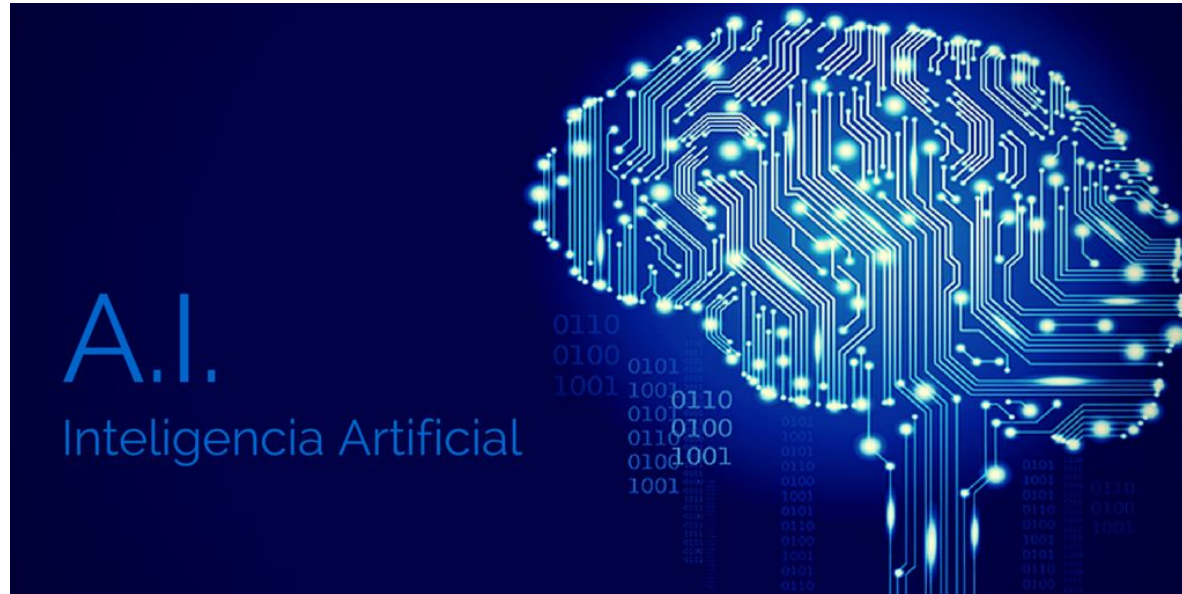
Los pasos son los siguientes:
Para comprarlo por internet:
a) Ir a la página...
b) Seleccionar la opción...
c) Ingresar tus datos...

*¿Cuál ChatBot parece
más humano y por qué?*



El Modelo Cognitivo

(*Pensar como humano*)



¿Qué es el modelo cognitivo?

Cognición proviene de la palabra “**conocer**” y es la capacidad de un ser vivo para **procesar información** proveniente de estímulos (Visuales, Auditivos, Sensoriales). El modelo contempla 4 etapas:

Actor / Etapa	Etapa 1: Sensación	Etapa 2: Memoria	Etapa 3: Pensamientos	Etapa 4: Resolución
Humano	Vista (El humano ve un objeto)	La información llega desde los ojos hasta el cerebro (Estímulos eléctricos)	El cerebro del humano procesa la información visual	El humano entiende que está viendo una manzana
IA	Se ingresa información al sistema (Matriz de píxeles de una imagen)	Los valores de la matriz son almacenados en una variable o estructura de información	Un algoritmo o modelo procesa la información	El modelo o algoritmo entiende que está viendo una manzana



¿ Cómo piensa un humano?

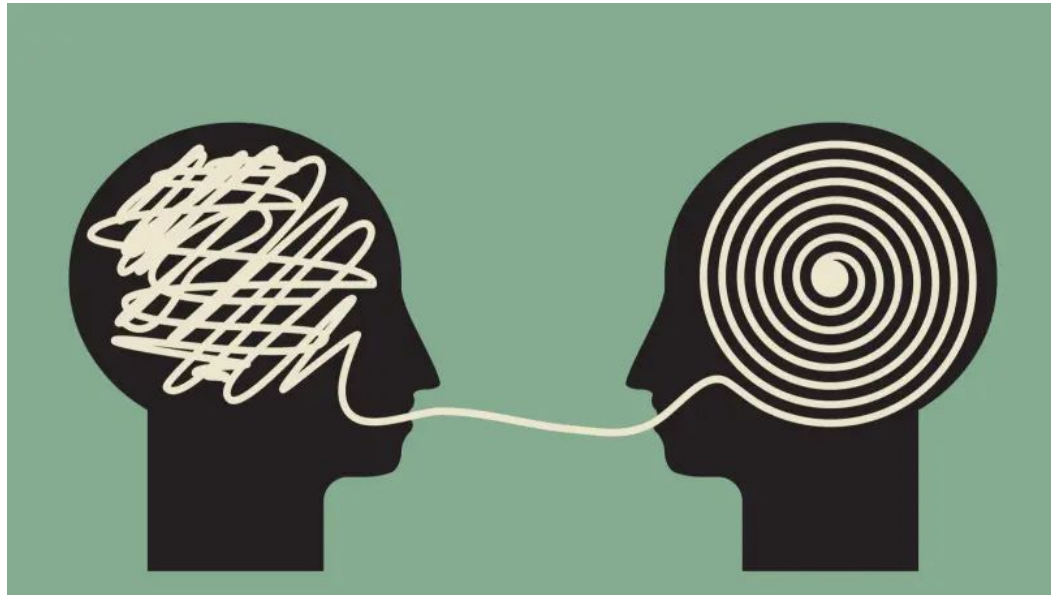
A ver, si eres tan “humano” ¿Cómo sabes que esto es una **manzana**?



Yo sé que esto es una manzana porque es / tiene:

Leyes del pensamiento

(*Pensar racionalmente*)



Leyes del pensamiento

Son un **conjunto de reglas** de **inferencia** que permiten llegar a una **consecuencia lógica válida**, y se basan en los 3 principios de la lógica:

Identidad: X es X

“Un perro es un perro, no es un gato, no es un caballo, es un perro”

No contradicción: Si es, no puede no ser

“Es de día o no es de día, pero no pueden ser ambos a la vez”

Tercero excluido: O es verdadero, o falso, pero no a medias

“Si una persona está viva, es verdadero que está con vida, y es falso que esté muerta, pero no hay un tercer juicio válido”

Ejercicio 1: Ley de Identidad

La ley de la identidad establece que un *objeto es idéntico a sí mismo*.

- Dada una lista de elementos (pueden ser números o cadenas de texto), escribe una función que cuente cuántas veces aparece cada elemento en la lista.
- La función debe devolver un diccionario donde las claves sean los elementos de la lista y los valores sean el número de veces que aparecen en la lista.
- No se permite el uso de librerías externas, solo estructuras de datos nativas de Python.

```
# Ejemplo de uso
lista = ["a", "b", "a", "c", "b", "a"]
resultado = contar_repeticiones(lista)
print(resultado)
```

Ejercicio 2: Ley de la NO contradicción

Establece que una proposición **NO puede** ser **verdadera y falsa al mismo tiempo**.

- Dada una lista de diccionarios donde cada diccionario representa una proposición con una clave y un valor booleano (True o False), escribe una función que verifique si existe alguna contradicción.
- Una contradicción ocurre si una misma clave tiene valores True y False en diferentes diccionarios.
- La función debe devolver True si hay una contradicción y False si no la hay.

```
# Ejemplo de uso
proposiciones = [
    {"A": True},
    {"B": False},
    {"A": False},
    {"C": True}
]

resultado = verificar_no_contradiccion(proposiciones)
print(resultado)
```

Ejercicio 3: Ley del tercero excluido

Establece que cualquier proposición **debe ser** o verdadera o falsa, **no existe un tercer estado**.

- Dada una lista que contiene valores booleanos (True o False) y otros posibles valores, escribe una función que identifique y devuelva los valores que no son estrictamente True o False.
- La función debe devolver una lista con los valores que no cumplen la ley del tercio excluido.
- No se permite el uso de librerías externas, solo estructuras de datos nativas de Python.

```
# Ejemplo de uso
lista = [True, False, 0, 1, "True", None, True]
resultado = identificar_valores_excluidos(lista)
print(resultado)
```