



UNIVERSIDAD
DE GRANADA



Inteligencia Artificial

Seminario 1: Sistemas inteligentes en un mundo abierto

E.T.S. de Ingenierías Informática y de Telecomunicación
Universidad de Granada

Este documento está protegido por la Ley de Propiedad Intelectual (Real Decreto Ley 1/1996 de 12 de abril).

Queda expresamente prohibido su uso o distribución sin autorización del autor.

Departamento de Ciencias de la
Computación e Inteligencia Artificial
<http://decsai.ugr.es>

Profesores de prácticas

- Juan Luis Suárez jlsuarezdiaz@decsai.ugr.es
 - Grupo 1: Aula 2.3, Jueves, 19:30
 - Tutorías: Jueves 17:30-19:30

- Nuria Rodríguez rbnuria@ugr.es
 - Grupo 2: Aula 3.6, Jueves, 17:30
 - Tutorías: Jueves 19:30-21:30

- Cristina Zuheros czuheros@ugr.es
 - Grupo 3: Aula 2.3, Viernes, 19:30
 - Tutorías: Martes 16:30-20:30
Miércoles 17:30-19:30



Evaluación de la parte de prácticas

En Convocatoria Ordinaria

Calificación de varias prácticas/pruebas (100%):

Práctica 1	Sistemas inteligentes en mundo abierto	20%
Práctica 2	Resolución de problemas con agentes reactivos/deliberativos	45%
Práctica 3	Resolución de un juego basado en técnicas de búsqueda	35%

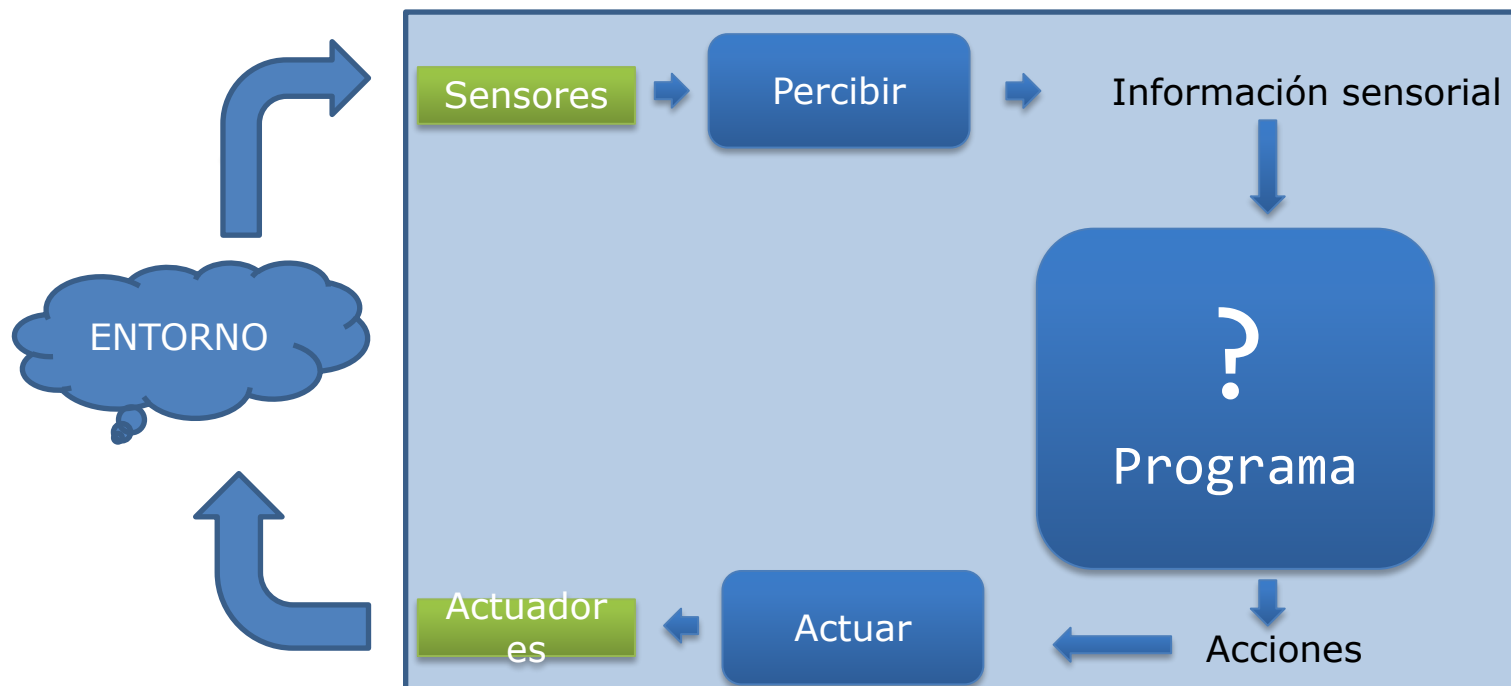
!!!No es obligatorio entregar todas las prácticas!!!

¡Es necesario alcanzar un 3 sobre 10 para hacer media con la teoría!

	Prácticas	Entregas
17-feb		
24-feb		
03-mar	Seminario P1	
10-mar	Resolución ejercicios Práctica	
17-mar	Seminario P2	17-03 Entrega P1
24-mar	Dudas y resolución relaciones de problemas	
31-mar	Dudas y resolución relaciones de problemas	
07-abr	Dudas y resolución relaciones de problemas	
14-abr	Semana santa	
21-abr	Dudas y resolución relaciones de problemas	
28-abr	Puente de mayo	04-05 Entrega P2
05-may	Seminario P3	
12-may	Dudas y resolución relaciones de problemas	
19-may	Dudas y resolución relaciones de problemas	
26-may	Dudas y resolución relaciones de problemas	30-05 Entrega P3

Objetivo del seminario: Un recorrido por los agentes del mundo cerrado, luego por los agentes del mundo abierto, terminando con la IA Generativa.

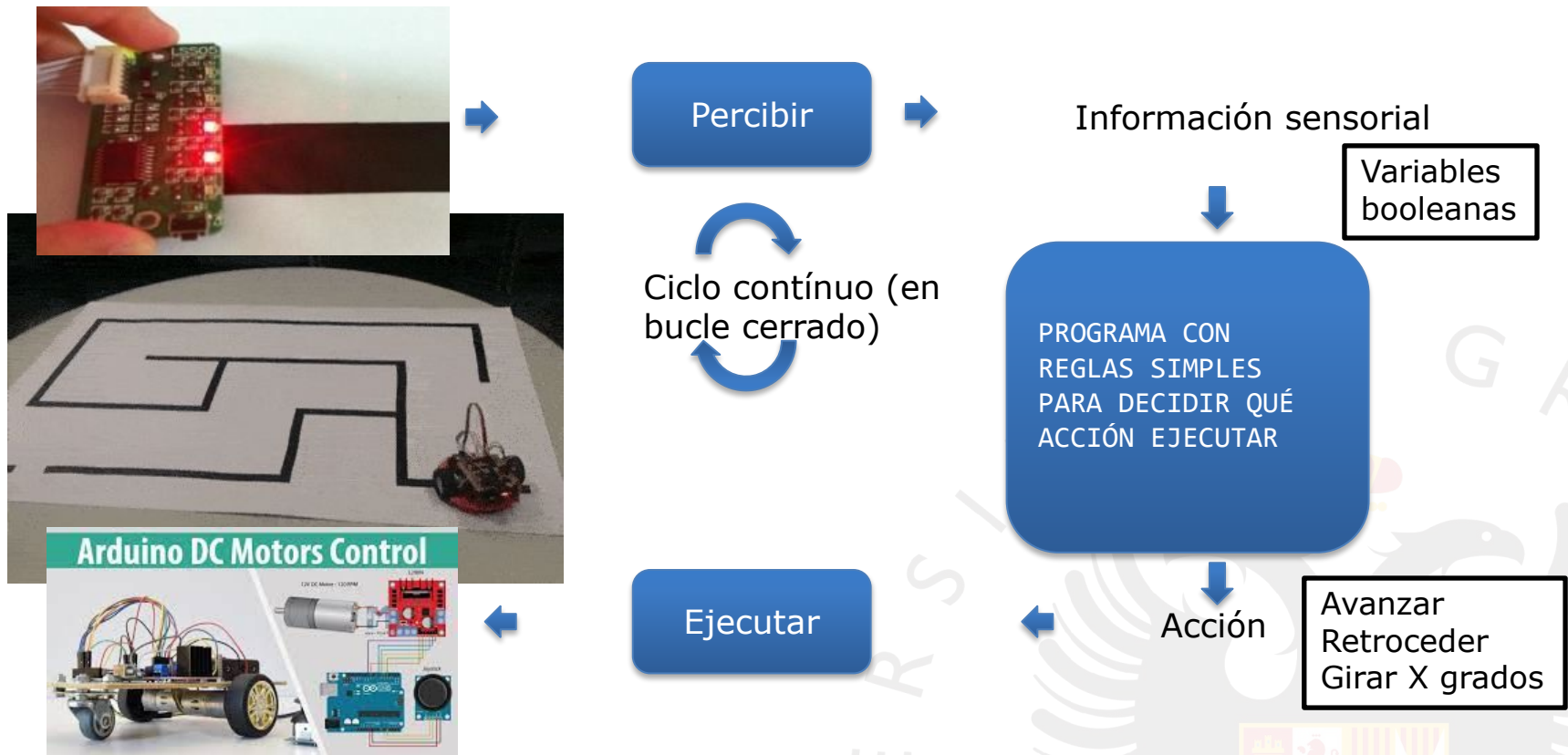
- Agentes (reactivos y deliberativos).
- Ejemplos con agentes deliberativos (8 puzle, planificación de caminos).
- Tipos de entornos en los que agentes desarrollan su comportamiento.
- Agentes en el mundo cerrado.
- Agentes en el mundo abierto.
- IA Generativa.
- Ejemplos con IA Generativa.



Un agente es un sistema que interactúa con un entorno a través de sensores y actuadores. Los sensores permiten al agente percibir información sensorial del entorno, la cual es procesada por un programa interno para tomar decisiones. Luego, mediante los actuadores, el agente ejecuta acciones que afectan su entorno. Este **ciclo continuo de percepción, procesamiento y acción** define el comportamiento del agente inteligente.

- Agentes reactivos
- Agentes deliberativos





Aplicación práctica de un agente reactivo diseñado para seguir una línea negra en un entorno controlado. El agente utiliza **sensores infrarrojos** para percibir la posición relativa de la línea, procesa esta **información sensorial y la representa internamente como variables booleanas**. Usa un **programa escrito** con (puede que muchas) **reglas simples** para decidir la acción más adecuada a la situación percibida y, finalmente, emplea **actuadores** como **motores controlados por una placa Arduino** para avanzar, retroceder o girar. Este sistema opera en tiempo real y **resuelve el problema de navegación** con reglas simples de percepción y acción.

Los agentes reactivos son adecuados para tareas de control, pero la Inteligencia Artificial pretende desarrollar sistemas que resuelven problemas o realizan tareas como lo hacen los humanos.



Vida cotidiana



Vida profesional



Vida ociosa

- Ejemplos de problemas que resolvemos los humanos:
 - Vida cotidiana: Planificación de rutas, Logística/transporte de niños en vehículos, usar varios transportes para ir de casa a la universidad.
 - Vida profesional: Transporte de objetos, Planificación de Tareas en el trabajo, Planificación de Tratamientos.
 - Vida ociosa: Resolver Puzles, Jugar al Ajedrez.

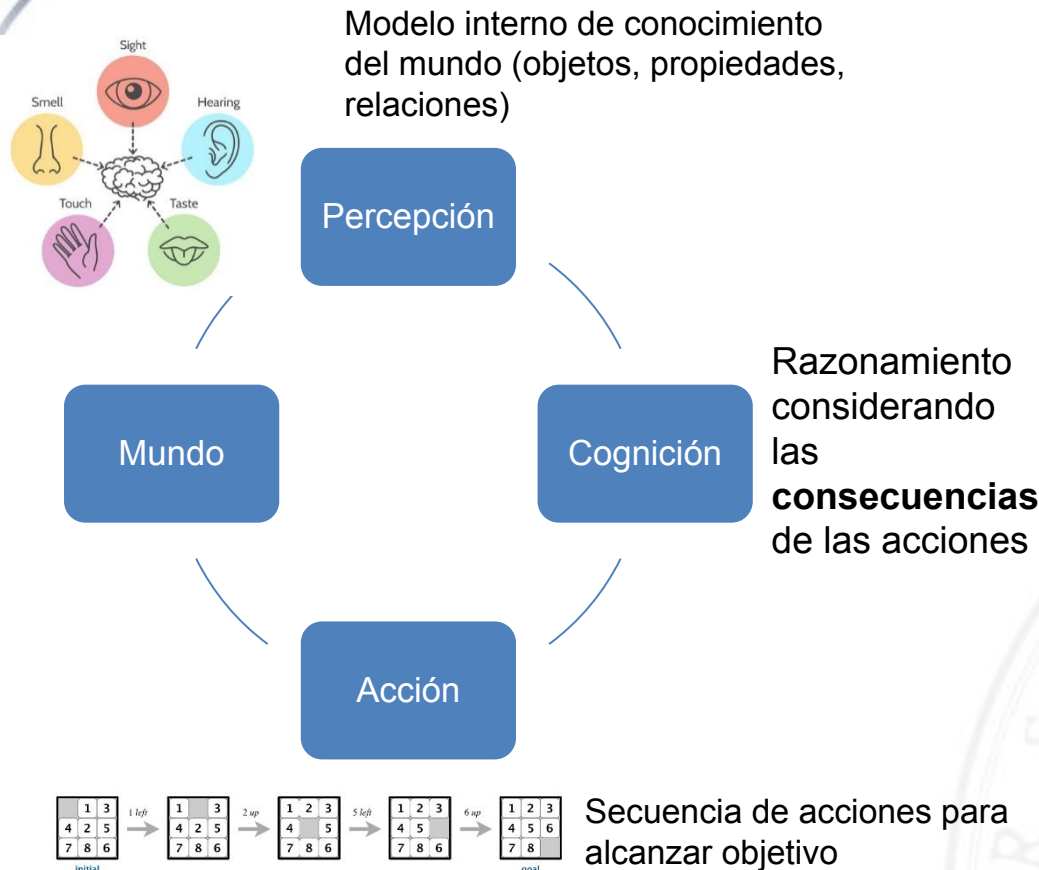
- Vamos a ver un ejemplo de cómo resolver el 8-puzzle

El **8-puzzle** es un juego de rompecabezas deslizante que consiste en un tablero de **3x3** con **8 fichas numeradas del 1 al 8** y una casilla vacía. El objetivo es **ordenar las fichas en un estado meta predefinido**, moviéndolas dentro del tablero utilizando la casilla vacía.

8		6
5	4	7
2	3	1

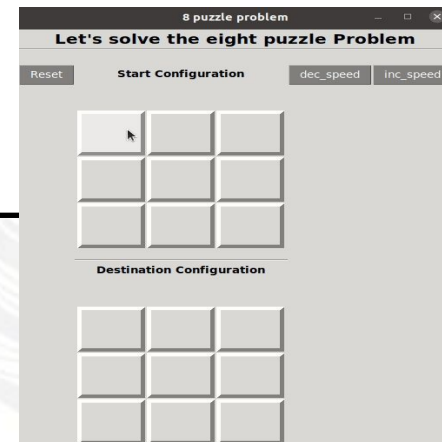


	1	2
3	4	5
6	7	8



Para resolver el 8 puzzle, los humanos empleamos un ciclo de procesos cognitivos de alto nivel que incluyen percepción (nos construimos un modelo interno del mundo), cognición (con el que razonamos) y acción (ejecutamos las acciones que hemos planificado).

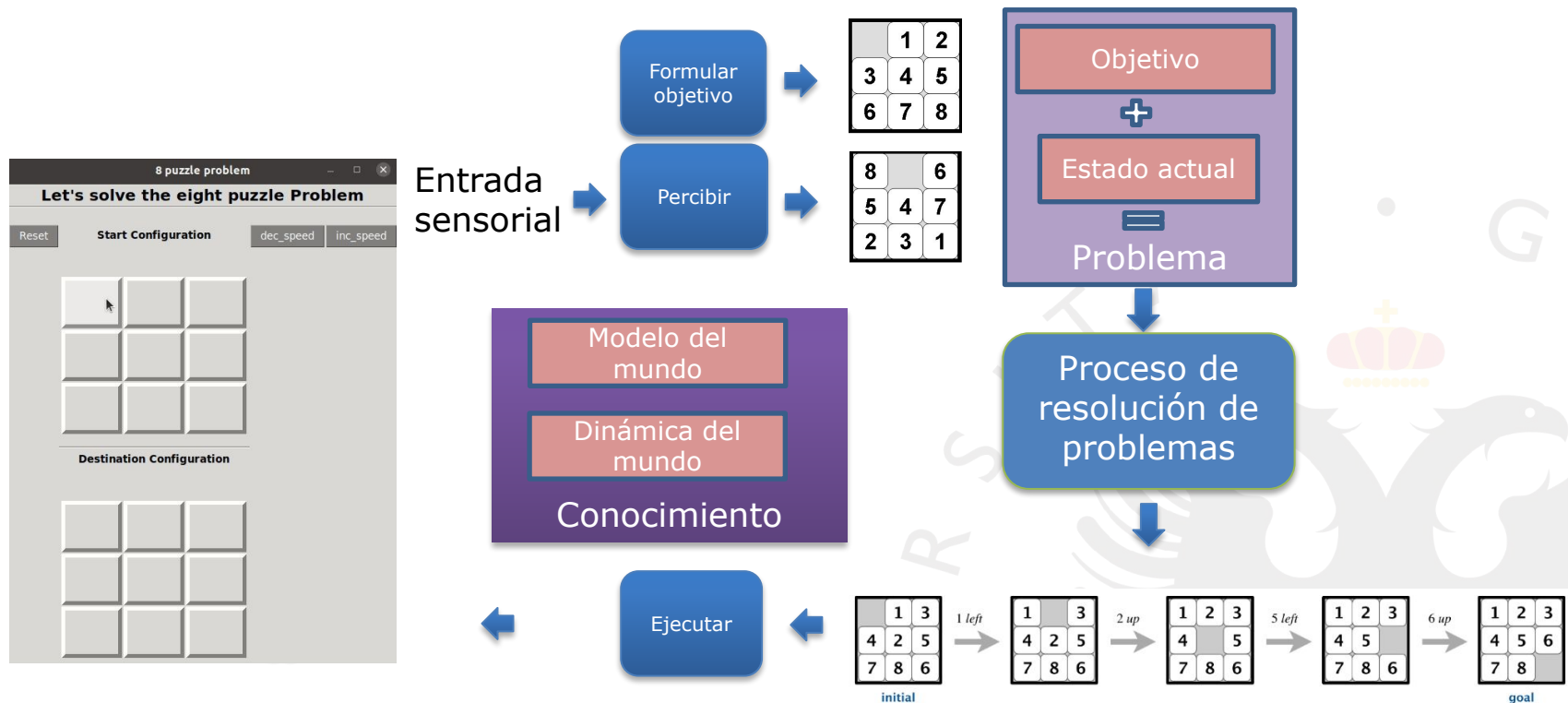
- ¿Y si queremos desarrollar un agente que resuelva el 8-puzzle?
 - Representar conocimiento del mundo.
 - Razonar sobre este conocimiento considerando como se transforma el mundo con las acciones.
 - Intención (objetivo, meta), **Planificar a largo plazo.**



Si pretendemos desarrollar un agente que resuelva el 8-puzzle tendremos que tratar de emular estos procesos

Conceptos esenciales de agente deliberativo (más detalles en Tema 3)

- Los agentes deliberativos son la propuesta de la IA para poder resolver este tipo de problemas. Están basados en el ciclo percepción, decisión y actuación de cualquier agente.
- **Orientados a objetivos:** además de percibir la situación o estado actual (una configuración inicial del 8 puzzle), **necesitan un objetivo** o situación final a alcanzar (configuración final del 8-puzzle).
- **Resolución de problemas:** proceso de decisión basado en **búsqueda** y **razonamiento**:
 - **Problema:** el agente recibe como entrada el problema a resolver, que consiste en una descripción del estado objetivo que se quiere alcanzar y del estado actual.
 - **Búsqueda:** Este proceso se basa en un algoritmo de búsqueda (que veremos en el tema 3)
 - **Razonamiento:** El algoritmo usa **conocimiento**
 - del mundo.
 - de la dinámica del mundo para encontrar una solución al problema.
- **Solución:** Una solución consiste en una secuencia de pasos (acciones) a ejecutar para pasar de la configuración inicial a la configuración objetivo.



8 puzzle: situar de forma ordenada, casillas numeradas del 1 al 8 desordenadas al inicio, moviendo la celda vacía

- Ir a la página <https://tristanpenman.com/demos/n-puzzle/>
- **Percepción:** Introducir el estado inicial y objetivo que se ve en las figuras a la derecha (esto es el problema)
- Seleccionar 'Búsqueda en anchura' (en detalle en el Tema 3)
- Seleccionar 'Single step' y ejecutar paso a paso.
- **Resolución del problema:** Clica en "Start"
 - **Observar:** El agente tiene como **representación del mundo** una matriz de números, que usa para representar estados.
- Clica en "Next"
 - **Observar:** Aplicando su conocimiento sobre la **dinámica del mundo** (movimientos válidos del 8-puzzle) genera en su memoria los estados consecuencia de la aplicación de los movimientos válidos posibles en el estado inicial. En rojo dibuja el siguiente que ha decidido explorar, en azul dibuja el estado que se deja pendiente de explorar.
- (seguir clicando en "Next") Se observa cómo el sistema expande nodos (razona sobre las consecuencias de sus acciones) y busca la solución.
 - **Observar:** El crecimiento exponencial en la memoria del agente
 - **Observar:** Encuentra el estado objetivo en 4 pasos,
- **Solución:** la solución al problema es el camino que va desde el estado inicial hasta el estado objetivo.
- La solución se encuentra en 4 pasos, pero explora muchos nodos.
 - **Observar:** explora muchos estados (nodos del espacio de búsqueda) hasta que encuentra la solución, ¿es posible explorar menos nodos?

Goal state:

1	2	3
4	5	6
7	8	-

Edit state

Initial state:

1	3	-
4	2	5
7	8	6

Edit state

¿Cómo hacer el proceso más eficiente?

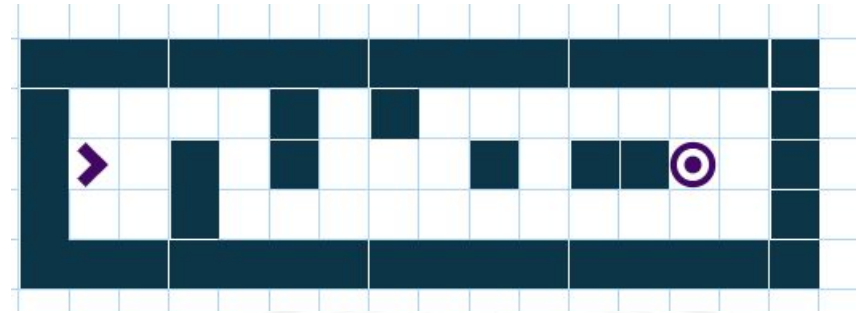
1. Reiniciar el proceso.
2. Seleccionar 'A* search' con heurística 'Tiles out-of-place' (en detalle en el tema 3).
3. Seleccionar 'Burst' para resolver de una vez.
4. **Observar:** Se encuentra la misma solución, pero explorando menos nodos.
5. Esto ha sido así porque ha utilizado **Información heurística** (más detalles en el tema 3)
 - En cada paso cuenta las casillas mal colocadas y decide explorar el siguiente estado que tiene un menor valor de casillas mal colocadas.

Initial state:			Goal state:		
1	3	-	1	2	3
4	2	5	4	5	6
7	8	6	7	8	-
Edit state			Edit state		

- Ventaja de los agentes deliberativos:
- Siempre que
 - Un problema pueda plantearse como una situación inicial y un objetivo
 - Y podamos escribir:
 - Una representación del mundo
 - Una descripción de la dinámica del mundo (cómo las acciones transforman el mundo)
 - Conocimiento heurístico para hacer eficiente el proceso de resolución
- Podremos usar un algoritmo de búsqueda, que razona a partir del conocimiento del mundo y su dinámica, para resolver cualquier problema (que tenga solución).

(veamos otro ejemplo)

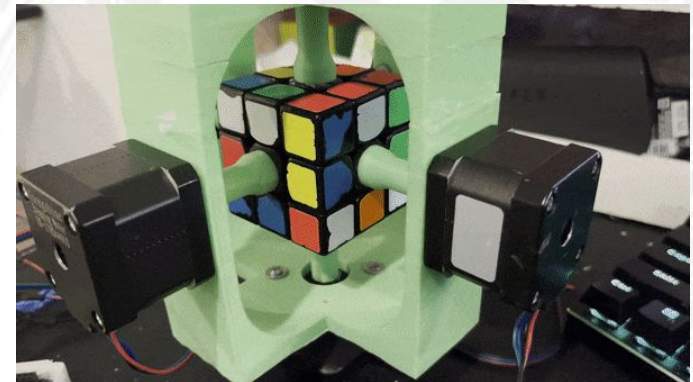
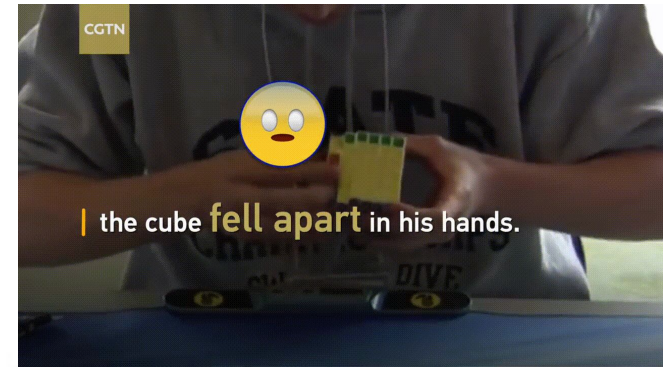
- Conocimiento:
 - Representación del mundo: un mapa con posiciones representadas como casillas.
 - Dinámica del mundo: descripción de los posibles movimientos en el mapa (por ejemplo, ir a izda, dcha, arriba, abajo)
 - Heurística: ante varias alternativas de movimiento elegir la que esté más cerca de la posición final (más detalle en el Tema 3)
- Problema:
 - Estado inicial: una posición cualquiera del mapa.
 - Inicial: posición (fila3, columna2)
 - Estado final: una posición destino en el mapa
 - Final: posición (fila3, columna12)
- Solución:
 - Pasos que hay que dar para ir desde la posición inicial hasta la posición final.



- Ir a la página <https://clementmihailescu.github.io/Pathfinding-Visualizer/#> (Cliclar en “Skip tutorial”)
- **Percepción:**
 - En el menú “Maze & Patterns”, seleccionar el mapa “Recursive Division (horizontal skew)”
 - La posición inicial y objetivo están ya marcadas.
 - **Observar:** El agente tiene como **representación del mundo** una matriz de casillas. Para representar estados basta con considerar la fila y columna de casilla que se está explorando
- **Resolución del problema sin heurística:**
 - En el menú “Algorithms” seleccionar el algoritmo “Breadth First Search”. Es un algoritmo que no usa heurística, como ya vimos en el ejemplo anterior.
 - En el menú “Speed” seleccionar “Fast”.
 - ¡Clicar en “Visualize BFS!”
 - **Observar:** en azul se muestran las posiciones que ha tenido que explorar antes de encontrar el camino. En amarillo muestra el camino más corto.
 - Tomar una captura de la pantalla, para comparar después.
- **Resolución del problema con heurística:**
 - En el menú “Algorithms” seleccionar el algoritmo “A* search”. Es un algoritmo que usa heurística, en cada paso hace una estimación de la distancia más corta al destino contando las filas y columnas (distancia Manhattan), no es información exacta, pero suficiente para guiar la búsqueda (más detalles en el tema 3)
 - **Observar:** comparar con la anterior captura de pantalla, hay menos casillas azules (nodos explorados) que en el proceso de resolución anterior (el proceso de búsqueda es más eficiente con heurística).
- **Solución:** la solución al problema es el camino que va desde el estado inicial hasta el estado objetivo.

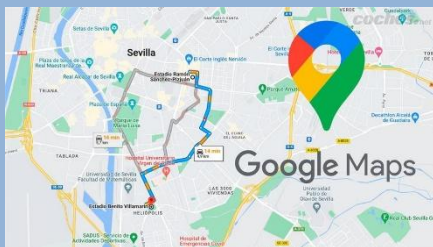
Y una vez que tenemos un plan, ¿qué hacemos?

- Los modelos de resolución de problemas que hemos visto pueden resolver problemas muy difíciles, y dar soluciones a un problema mediante un plan.
- Pero a la hora de ejecutar ese plan en el mundo real, pueden pasar muchas cosas que den al traste con el mismo.
 - Google maps usa algoritmos de planificación de caminos.
 - ¿Qué pasa si mientras conducimos la ruta que nos ha dicho tiene una carretera cortada?
- Supongamos que queremos resolver el cubo de Rubik. Depende de cómo sea el entorno, el cubo de Rubik puede tener o no problemas de ejecución:
 - Si asumimos que los actuadores no fallaron, que la percepción es siempre correcta y que nuestro algoritmo es fiable, podemos usar agentes como los que acabamos de ver.
 - Pero si lo ejecutamos en el mundo real pueden pasar muchas cosas.
- Vamos a ver algunas características de los entornos.



Característica		
Determinismo (relativo a las acciones)	Entorno Determinista (Mover pieza en ajedrez)	Entorno no determinista (Golpear balón)
Observabilidad (relativo a los sensores)	Observabilidad total (Hay sensores para todo aspecto relevante para decidir y actuar)	Observabilidad parcial (No dispones de algún sensor o los sensores tienen ruido/imprecisión)
Dinamismo del mundo	Entorno estático (el mundo solo lo cambia el agente, control absoluto del entorno)	Entorno dinámico (en el mundo hay otros agentes o eventos que lo cambian fuera del control del agente)
Compleitud del conocimiento	Conocimiento completo (Se conocen todas las leyes del mundo)	Conocimiento incompleto (O no se conoce ninguna ley del mundo o algunas parcialmente)
Continuidad	Entorno discreto (conjunto finito de estados o acciones en intervalos discretos de tiempo, p.ej Videojuegos en mundos cuadrícula)	Estados y acciones continuas (p.ej. Coche autónomo: Velocidad, posición, ángulo de giro, velocidad de giro)

- Los ejemplos de agentes anteriores son de agentes en el mundo cerrado.
- Están diseñados para operar en entornos
 - Totalmente observables, discretos, estáticos, totalmente conocidos
- Necesitan tener una representación del conocimiento
 - Conocimiento del mundo, de la dinámica del mundo, heurísticas.
 - En general, escrita por humanos.
 - En algunos casos pueden aprender cómo mejorar la eficiencia, aprendiendo heurísticas como en el ajedrez.
 - Por ejemplo, AlphaGo superó al campeón mundial de Go en 2017 porque aprendió a partir de (millones) de partidas previamente conocidas (el entorno es completamente conocido).
- La IA ha llegado a grandes logros con agentes del mundo cerrado.



Planificar rutas



Juegos bipersonales

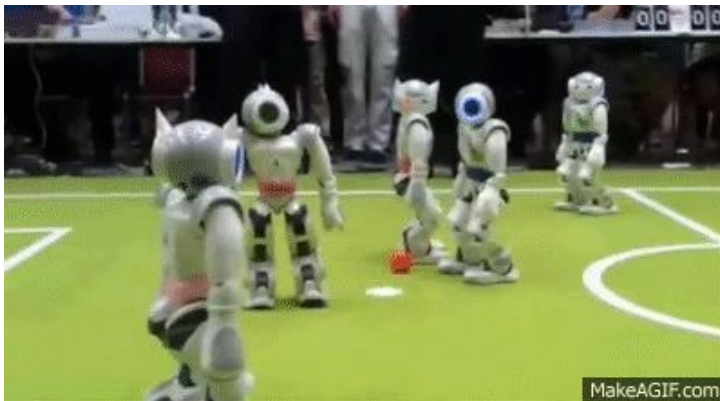


¿Los entornos más difíciles para un agente?

- Son los entornos del mundo abierto, en general
 - No determinista,
 - Parcialmente observable,
 - Dinámico,
 - Continuo,
 - (parcialmente)Desconocido.



- **No determinista:** El resultado de una acción (golpear la pelota) no es siempre el mismo.
- **Parcialmente observable:** Los robots no pueden ver todo el campo al mismo tiempo; ni obtener información completa sobre la posición del balón y los adversarios.
- **Dinámico:** El entorno cambia constantemente debido a los movimientos de los jugadores y la trayectoria impredecible del balón.
- **Continuo:** El estado del juego no se puede representar en un número finito de valores discretos.
- **Desconocido:** Antes del partido, los robots no tienen conocimiento detallado del comportamiento de los oponentes, las condiciones exactas del entorno o posibles fallos en sensores y actuadores.



<https://2024.robocup.org/>



- Para enfrentarse a estos problemas los agentes de IA deben tener la capacidad de:
 - **adaptarse** a situaciones imprevistas o nuevas (es decir, saber responder de forma fiable ante estas situaciones),
 - interactuar y **aprender del entorno** (posiblemente en tiempo real), haciendo frente a información parcial o ambigua (por ejemplo, observar en el entorno que una decisión o acción es errónea o no adecuada, y aprender cómo responder),
 - **generalizar** su conocimiento y experiencia para enfrentar desafíos que no estaban previstos inicialmente (poder llevar a cabo nuevas tareas, sin reprogramar, a partir de experiencia previa).

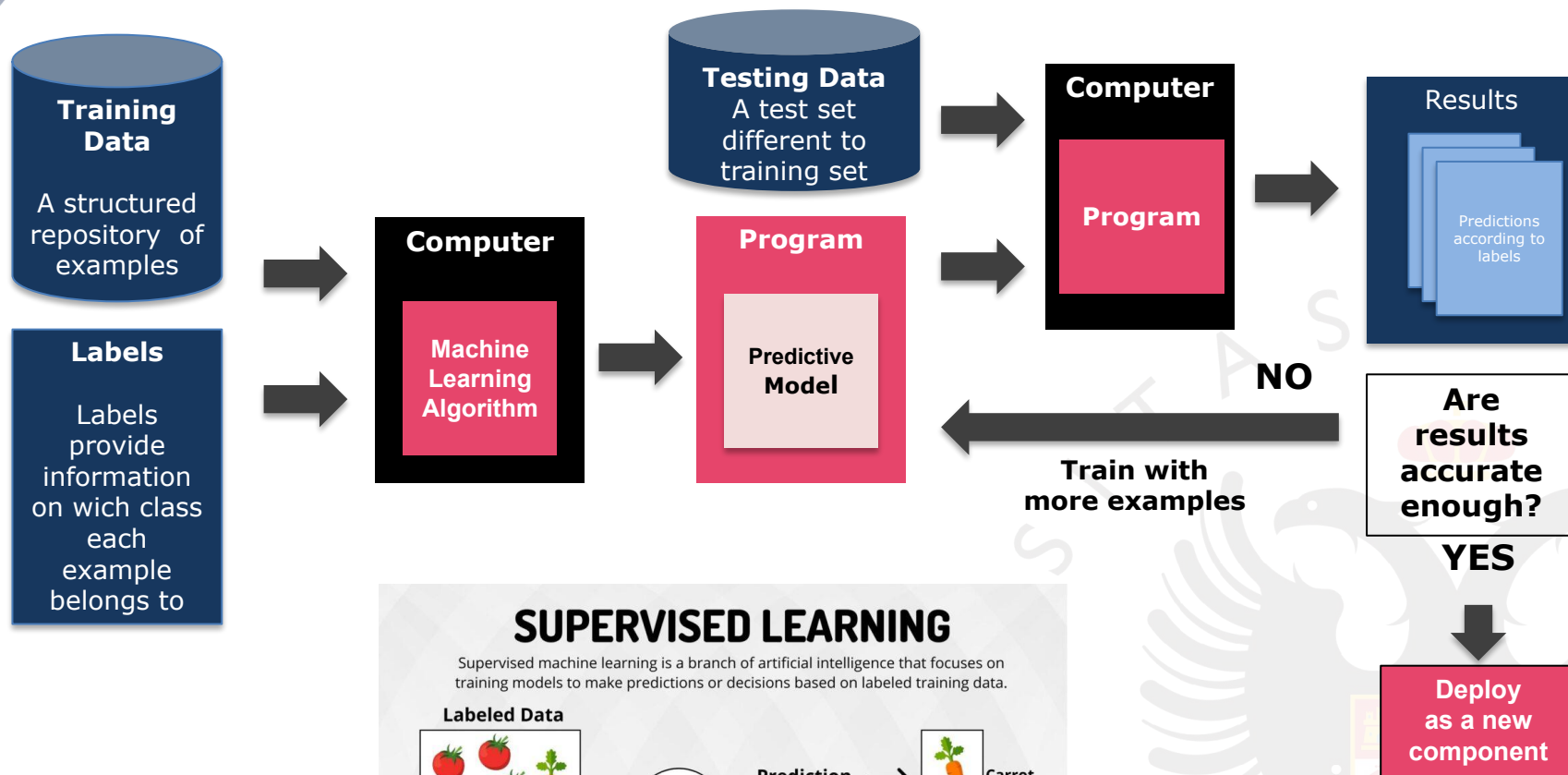
Un momento, ¿en qué consiste aprender?

- El **aprendizaje automático (machine learning)** (en detalle en el Tema 6) es una rama de la inteligencia artificial que permite a las computadoras aprender a realizar tareas sin ser programadas explícitamente para cada caso. En lugar de escribir reglas específicas, se entrenan modelos usando datos, de manera que el sistema reconoce patrones y mejora su rendimiento con la experiencia.
- **¿Cómo se lleva a cabo? (esquema en la siguiente diapositiva)**
 1. **Recopilación de datos:** Se obtienen datos relevantes para la tarea que queremos automatizar.
 2. **Entrenamiento del modelo:** Se usa un algoritmo de aprendizaje que analiza los datos y encuentra patrones.
 3. **Evaluación y ajuste:** Se prueba el modelo con datos nuevos para ver cómo de bien predice o clasifica, ajustando sus parámetros si es necesario.
 4. **Uso en el mundo real:** Una vez entrenado, el modelo puede hacer predicciones o tomar decisiones de forma autónoma.
- *Ejemplo: Un modelo de aprendizaje automático puede aprender a reconocer imágenes de gatos tras analizar miles de fotos etiquetadas. Luego, cuando se le muestra una imagen nueva, puede determinar si hay un gato en ella o no.*

Step1: Training

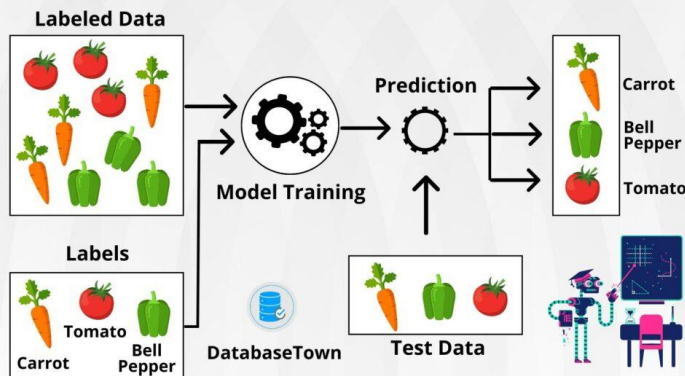
Step2: Testing

Step3: Deploy



SUPERVISED LEARNING

Supervised machine learning is a branch of artificial intelligence that focuses on training models to make predictions or decisions based on labeled training data.



- Agentes **Autónomos** en Open-World: operan sin intervención humana directa, aprendiendo del entorno y tomando decisiones de forma independiente.
 - Mundo físico: coches autónomos.
 - Mundo virtual: agentes en open-world games.
- Agentes **Colaborativos** en Open-World: interactúan con humanos en entornos abiertos para facilitar tareas.
 - Mundo físico: Robots asistenciales.
 - <https://sites.google.com/correo.unimet.edu.ve/robotica-paradiscapacitados/robots-asistenciales>
 - Ver ROBEAR.
 - Mundo virtual: IA generativa (ChatGPT, Copilot, DALL·E), que colabora con humanos en creatividad, escritura y programación.

Autonomía en mundo abierto físico

- ✓ 1. Se adapta a situaciones imprevistas**
 El coche autónomo **detecta y responde** a eventos inesperados como un peatón cruzando, un coche detenido o lluvia intensa. No sigue rutas fijas, sino que **recalcula y ajusta su conducción en tiempo real**.
 - Ejemplo:* Si hay un accidente bloqueando el camino, **busca rutas alternativas y ajusta la velocidad**.
- ✓ 2. Aprende interactuando con el entorno**
 Utiliza datos pasados (**aprendizaje offline**) y experiencias en tiempo real (**aprendizaje online**) para mejorar su conducción.
 - Ejemplo:* Si detecta tráfico en una calle con obras, **recordará evitarla en futuras rutas**.
- ✓ 3. Generaliza su experiencia a nuevos desafíos**
 Aplica lo aprendido en un entorno a otros diferentes sin necesidad de reentrenamiento específico.
 - Ejemplo:* Si ha aprendido a reconocer señales de tráfico en EE.UU., podrá **identificar señales similares en otros países**.



<https://www.agenciasinc.es/Noticias/CARLA-un-entorno-virtual-para-entrenar-a-los-coches-autonomos>

Autonomía en mundo abierto virtual

- En *Minecraft* se han desarrollado agentes que no solo siguen instrucciones predefinidas, sino que **aprenden, se adaptan y exploran por sí mismos**.
- *Odyssey* y *Jarvis*, sistemas basados en IA que convierte a un agente en un **open-world agent**, capaz de enfrentarse a situaciones nuevas e imprevistas.
- ☒ **1. Se adapta a lo inesperado**
Si el agente necesita diamantes, pero no tiene un pico de hierro, no se queda bloqueado. **Detecta el problema, fabrica primero el pico y luego sigue con su objetivo.**
- ☒ **2. Aprende interactuando con el entorno**
Cada vez que el agente toma una decisión, recibe retroalimentación. **Si muere por un zombi sin armas, aprenderá que la próxima vez debe llevar una espada y armadura antes de explorar.**
- ☒ **3. Generaliza su experiencia para nuevos retos**
Si aprende a construir un refugio en un bosque, podrá aplicar ese conocimiento en un desierto o en la nieve, **ajustando su estrategia según el entorno.**



<https://craftjarvis-jarvis1.github.io/>

En este enlace hay una demo interactiva de cómo mejorar el comportamiento para una misma tarea

<https://github.com/zju-vipa/Odyssey>



Colaboración en mundo abierto físico

- ✓ **1. Se adapta a situaciones imprevistas**
 Robear, un robot asistencial, **ajusta su comportamiento** según las necesidades del paciente y el entorno, adaptándose a **cambios en la postura, resistencia inesperada o movimientos involuntarios**.
 ♦ *Ejemplo:* Si un paciente pierde el equilibrio al intentar levantarse, **Robear detecta el problema y ajusta su agarre para estabilizarlo**.
- ✓ **2. Aprende interactuando con el entorno**
 Usa **sensores y experiencia previa** para mejorar su capacidad de asistencia, ajustando la fuerza y el ángulo de elevación según cada usuario.
 ♦ *Ejemplo:* Si un paciente prefiere cierta inclinación al ser levantado, **Robear recordará esa preferencia y la aplicará en futuras interacciones**.
- ✓ **3. Generaliza su experiencia a nuevos desafíos**
 No necesita ser reprogramado para cada persona; aprende patrones y **los aplica a distintos usuarios y situaciones**.
 ♦ *Ejemplo:* Si ha aprendido a **levantar a una persona con movilidad reducida**, podrá **asistir a otro usuario con diferente peso o altura** sin necesidad de un ajuste manual.



<https://sites.google.com/correo.unimet.edu.ve/roboticaparadisdiscapacitados/robots-asistenciales>

Colaboración en mundo abierto virtual

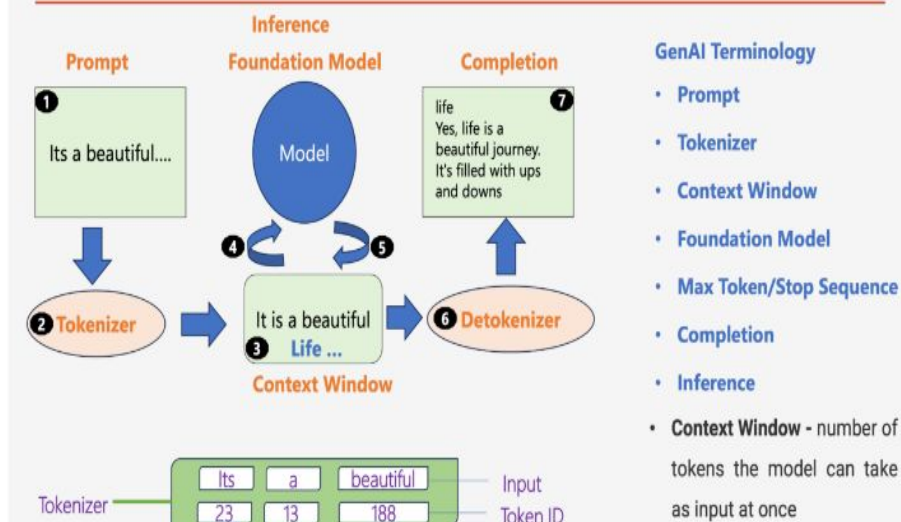
- ✓ 1. Se adapta a situaciones imprevistas**
 La IA generativa responde **dinámicamente a diferentes solicitudes**, adaptando su salida según el contexto y las necesidades del usuario.
 - Ejemplo:* Si un usuario pide un texto formal pero luego lo quiere más coloquial, **la IA ajusta el tono sin necesidad de reescribir todo desde cero.**
- ✓ 2. Aprende interactuando con el entorno**
 Puede refinar sus respuestas basándose en el historial de interacción y el feedback del usuario. Además, en algunos casos, **se actualiza con nuevos datos para mejorar su desempeño.**
 - Ejemplo:* Un asistente de escritura basado en IA mejora sus sugerencias a medida que el usuario **acepta o corrige palabras y frases.**
- ✓ 3. Generaliza su experiencia a nuevos desafíos**
 No está restringida a tareas específicas, sino que **aplica su conocimiento en múltiples áreas**, desde generación de imágenes hasta asistencia en programación o redacción.
 - Ejemplo:* Si ha aprendido a escribir un ensayo sobre historia, también puede **generar artículos científicos o guiones de cine aplicando estructuras similares.**



<https://opencv.org/blog/what-is-generative-ai/>

- La Inteligencia **Artificial Generativa (GenAI)** se refiere a técnicas computacionales basadas en aprendizaje automático que tienen la capacidad de crear contenido nuevo y significativo, como texto, imágenes o audio, a partir de datos de entrenamiento.
- Está basada en modelos generativos de aprendizaje automático. Estos modelos se entrenan con grandes volúmenes de datos y pueden producir respuestas que imitan el lenguaje humano o generar contenido en diversas modalidades.
- Se estima que GPT-4 fue entrenado con un conjunto de datos significativamente mayor que su predecesor, GPT-3, que utilizó “la huella” del conocimiento y la cultura humana disponible en internet con una amplia variedad de fuentes de datos extraídas de la web. Además, se estima que GPT-4 cuenta con alrededor de 1,76 billones de parámetros, en comparación con los 175 mil millones de parámetros de GPT-3.
- <https://www.ucl.ac.uk/teaching-learning/generative-ai-hub/introduction-generative-ai#What%20is%20GenAI?>

How Generative AI works - Basic Concepts and Terminology - 2



- **Prompt (texto de entrada):** Se introduce una frase de entrada ("It's a beautiful...").
- **Tokenización:** El prompt se convierte en **tokens** numéricos que el modelo puede procesar.
- **Contexto (Context Window):** Número máximo de tokens que el modelo puede manejar simultáneamente para determinar la semántica del texto.
- **Foundation Model:** El modelo de IA procesa la entrada y predice la siguiente palabra o secuencia.
- **Generación:** El modelo genera una continuación del texto basada en los tokens previos de contexto.
- **Detokenización:** La salida generada en tokens se convierte de nuevo en texto.
- **Completion:** Se obtiene el resultado final con una frase completa ("Life... Yes, life is a beautiful journey...").

Términos que se usan con frecuencia en las descripciones de GenAI, no es necesario comprender exactamente qué significan todos. Lo más importante es notar la jerarquía de las tecnologías y su complejidad.

<https://www.ucl.ac.uk/teaching-learning/generative-ai-hub/introduction-generative-ai/#What%20is%20GenAI?>

ChatGPT (y otras GenAI de texto) es una aplicación basada en:

- Generative Pre-trained Transformer (GPT: un tipo avanzado de LLM)
- Modelo de Lenguaje Grande (LLM: una representación computarizada masiva de ejemplos de lenguaje natural) <https://bbycroft.net/llm>
 - Modelo Fundacional: entrenado con texto, imágenes y/o sonido.
 - <https://serokell.io/blog/guide-to-foundation-models>
- Transformer de propósito general (un procesador de lenguaje basado en RNA) <https://poloclub.github.io/transformer-explainer/>
- Red Neuronal Artificial (RNA: un enfoque de aprendizaje automático -ML inspirado en el cerebro humano y sus conexiones sinápticas) <https://playground.tensorflow.org/>
- Aprendizaje Automático (ML: un enfoque de IA que usa algoritmos para mejorar automáticamente con datos) <https://ml-playground.com/>

Fortalezas

- Puede producir contenido diverso y aparentemente original.
- Es capaz de procesar e interpretar lenguaje humano.
- Puede generar texto en múltiples idiomas.
- Se puede ajustar para diferentes tareas y dominios.
- Aprende patrones a partir de grandes volúmenes de datos.
- Puede recordar interacciones previas, mejorando la coherencia.
- Cuando genera respuestas rápidamente, ideal para aplicaciones en tiempo real.

Debilidades

- Puede generar información inexacta o engañosa.
- No comprende realmente el contenido que genera.
- Puede infringir derechos de propiedad intelectual.
- Puede producir contenido éticamente problemático.
- Requiere altos recursos computacionales y energéticos.
- Su contenido necesita moderación para evitar sesgos y daños.
- Puede ser utilizada para generar noticias falsas y deep fakes.
- Puede reforzar la brecha digital al favorecer a grandes empresas.
- Su funcionamiento es opaco, dificultando la trazabilidad de sus respuestas.
- Puede inundar Internet con contenido generado, afectando futuros modelos.

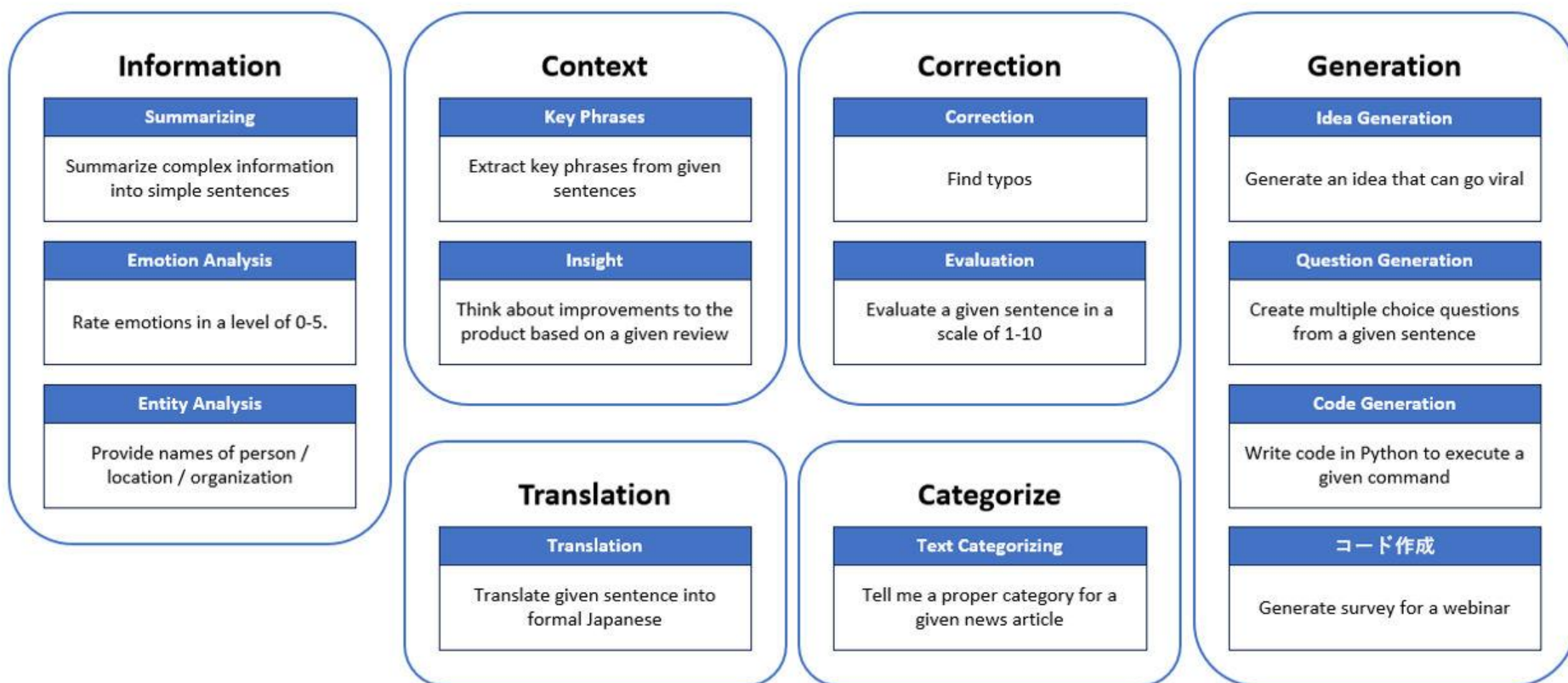
- **Deep Learning:** Avance tecnológico en el aprendizaje automático.
- **Computación con GPUs:** Aumento exponencial de la capacidad de cómputo.
- **Big Data:** Acceso a gran cantidad de datos con los que entrenar los modelos.
- **Inversión:** Enorme inversión por parte de grandes empresas y actores que han visto una oportunidad de mercado muy relevante.

 / [Formacion tic](#) / [Inteligencia artificial](#) / RECOMENDACIONES IA

Recomendaciones para usar la IA en la UGR

<https://ceprud.ugr.es/formacion-tic/inteligencia-artificial/recomendaciones-ia>

La GenAI no está destinada a resolver problemas como hacemos los humanos, sino a colaborar con los humanos en tareas cotidianas de procesamiento de información.



<https://us.kddi.com/en/resources/knowledge/solutionblog-20231027/>

- Con la cuenta <usuario>@go.ugr.es
 - se puede acceder a Gemini <https://gemini.google.com/app>
 - Se puede crear una cuenta en DeepSeek <https://chat.deepseek.com/>
- Podemos crear una cuenta ChatGPTFree en <https://chatgpt.com/>
 - Por ejemplo, con un nuevo correo de Google
- Recomendación: hacer grupos de 3 y que cada uno use un chatbot diferente.

La **IA Generativa** puede resumir textos extrayendo ideas clave y contexto, generando versiones concisas de documentos extensos. Es eficaz para simplificar contenido complejo, aunque requiere verificación humana para garantizar fidelidad y evitar omisiones críticas.

- Prompt: Genera una tabla comparativa con las principales características y diferencias entre ChatGPT, DeepSeek y Gemini
- Prompt: resume brevemente los principales algoritmos de búsqueda en espacio de estados del libro de Russel & Norvig para un estudiante de informática que aun no los conoce, y los va a conocer en breve
- Descargar el artículo <https://arxiv.org/abs/2306.02781> titulado "A survey of Generative AI Applications" y pedir que lo resuma.
- Discutir el concepto de "prompt engineering",
 - elaboración de prompts adecuados mensajes claros, concisos y específicos, evitando ambigüedades y proporcionando suficiente contexto,
 - para obtener los resultados esperados.

La **IA Generativa** puede generar código funcional interpretando instrucciones en lenguaje natural, identificando patrones y estructuras de programación aprendidas. Es útil para automatizar tareas repetitivas o prototipos rápidos, pero requiere validación humana para garantizar precisión y optimización.

- Prompt: genera un programa que pueda recibir como entrada una configuración del 8 puzzle como una cadena de números, por ejemplo 130425786 y de como salida los posibles movimientos válidos para esa configuración. Además, los posibles estados sucesores a la configuración de entrada. Además, que muestre la salida como una cuadrícula
- Después, comprobar que es correcto.

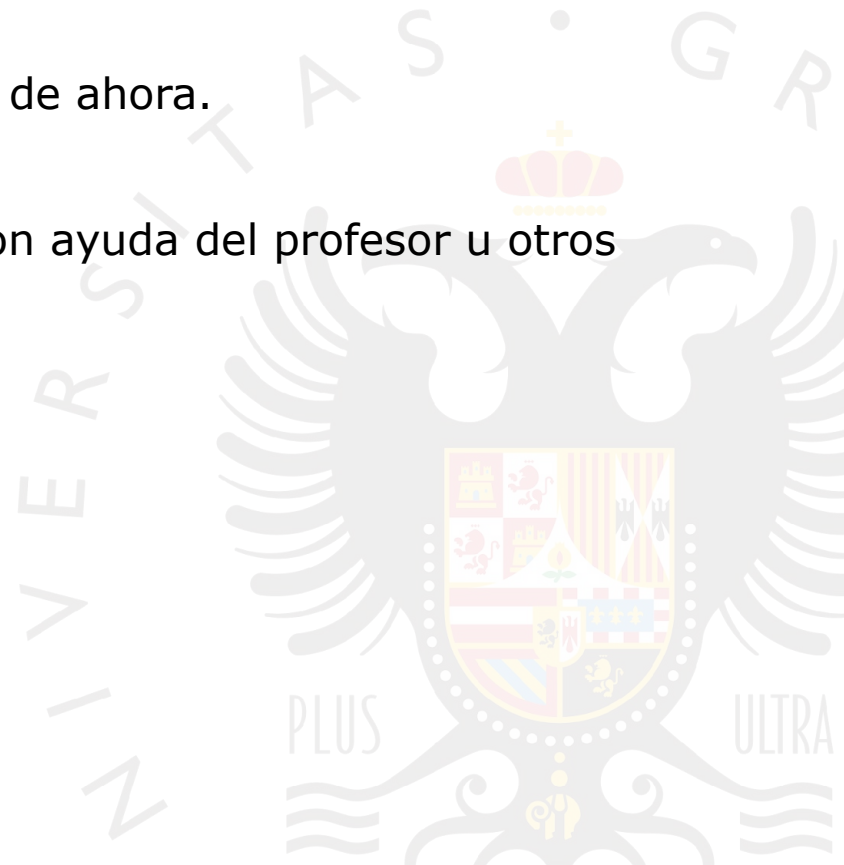
La **IA Generativa** puede simular procesos de razonamiento analizando datos y aplicando patrones aprendidos, útil para resolver problemas o deducciones básicas. Sin embargo, carece de comprensión real y puede cometer errores en contextos complejos, necesitando supervisión humana para validar resultados.

- Los modelos de IA Generativa de última generación (ChatGPT4, DeepSeek) están entrenados también para llevar a cabo tareas de razonamiento.
- No vamos a definir ahora en qué consiste el razonamiento, pero parece sensato pensar que razonar es lo que hacen los agentes deliberativos, que necesitan una representación del mundo, su dinámica y heurísticas.
- Los modelos de IAGen no tienen tal representación, pero han sido reajustados con diversas técnicas para simular razonamiento.
- Ejemplo (probar la clase con los tres chatbots cada alumno usa un chatbot)
 - Prompt1: Genera una configuración inicial, aleatoria, del 8 puzle y visualízala en modo texto
 - Prompt2: ¿cuáles son los movimientos posibles a partir de esta configuración? Dibújalos y No generes código, solo visualiza en texto
 - Prompt3: <decirle que> No es correcto
 - NOTA: el único que falla es Gemini.
- Discutir en clase las diferentes respuestas de cada chatbot.
 - Prompt4: a partir de esta configuración de 8 puzle: 130425786, y considerando que el puzle resuelto es 123456780, encuentra la solución, muéstrala según el código R,L,D,U para representar que la celda vacía se mueve a la derecha, izquierda, abajo y arriba, respectivamente
Nota: la solución es L D R D, que es la que hemos encontrado con el ejemplo anterior
- Discutir en clase las diferentes respuestas de los ChatBots.

Con la IAGen se puede solicitar que represente la respuesta en un formato conocido y reutilizarlo en otras aplicaciones.

- **Prompt:** genera un pseudocódigo de un algoritmo general de búsqueda en espacio de estado, formatealo en una estructura clara y bien organizada para su fácil interpretación.
- **Prompt2:** Usa el formato PlantUML para dibujar el proceso con carriles.
- Copiar la respuesta y pegarla en una aplicación online compatible con este formato, por ejemplo <https://www.planttext.com/>

- Entrega: dos ejercicios que consisten en cuestionarios sobre preguntas y ejercicios relacionados con los conceptos vistos en este seminario.
- La calificación será la media ponderada de las calificaciones de cada ejercicio. El ejercicio 1 tiene una ponderación del 30% y el ejercicio 2 del 70%.
- El cuestionario se puede rellenar a partir de ahora.
- En la siguiente sesión podrá realizarse con ayuda del profesor u otros compañeros.



Fecha límite entrega

17 de Marzo hasta las 23 horas

Esta práctica es INDIVIDUAL

En el caso de detectar prácticas copiadas, los involucrados (tanto el que se copió como el que se ha dejado copiar) tendrán suspensa la asignatura.

Ante cualquier sospecha de:

- Copia con otros estudiantes
- Copia de repositorios de internet
- Copia de trabajos de otros años
- El trabajo ha sido realizado por otra persona

Se podrá convocar a **defensa de prácticas** a los implicados. La defensa consistirá en una **prueba oral y/o escrita** con preguntas acerca del trabajo desarrollado en la práctica. Se realizará en las **horas de clase**.

La nota de la práctica quedará condicionada a la defensa.

Si no se supera la defensa, la calificación de la práctica será 0.

Si se confirman sospechas de copia, la convocatoria quedará automáticamente suspensa.