



XUNTA
DE GALICIA

CENTRO DE
FORMACIÓN E
RECURSOS DE FERROL

Inteligencia Artificial para la Sociedad



Alma Mallo (UDC)

Francisco Bellas (UDC)

Noviembre de 2024



Organización del curso

1. Sesión 1: introducción a la IA y aspectos ético-legales
2. Sesión 2: percepción y actuación
3. **Sesión 3: representación y razonamiento**
4. Sesión 4: aprendizaje automático 1
5. Sesión 5: aprendizaje automático 2
6. Sesión 6: herramientas de IA para la educación (IA generativa)



Sesión 3.

Representación y razonamiento



Representación y razonamiento

- Los agentes mantienen representaciones del mundo y las utilizan para el razonamiento





Representación digital

- El **sistema binario** únicamente emplea ceros y unos, que representan los estados “**apagado**” y “**encendido**”.
- En el sistema binario, cada dígito se denomina bit y solamente podrá adoptar como **valores 0 o 1**.





Representación digital

- El sistema hexadecimal emplea base 16.
- Dígitos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
- En comparación con otras representaciones, se trata de un sistema más compacto.

| | | |
|------------------|-------------------|-------------------|
| 0 0 0 0 | 0 | 0 0 |
| 0 0 0 1 | 1 | 0 1 |
| 0 0 1 0 | 2 | 0 2 |
| 0 0 1 1 | 3 | 0 3 |
| 0 1 0 0 | 4 | 0 4 |
| 0 1 0 1 | 5 | 0 5 |
| 0 1 1 0 | 6 | 0 6 |
| 0 1 1 1 | 7 | 0 7 |
| 1 0 0 0 | 8 | 0 8 |
| 1 0 0 1 | 9 | 0 9 |
| 1 0 1 0 | A | 1 0 |
| 1 0 1 1 | B | 1 1 |
| 1 1 0 0 | C | 1 2 |
| 1 1 0 1 | D | 1 3 |
| 1 1 1 0 | E | 1 4 |
| 1 1 1 1 | F | 1 5 |
| (x) ₂ | (x) ₁₆ | (x) ₁₀ |

- ¿Cómo representamos textos?
 - Asignando números a los caracteres.
 - Ese proceso se conoce como codificación de caracteres.
 - Existen distintos esquemas de codificación, los más conocidos son: ASCII, ASCII extendido y Unicode.

| Caracteres de control ASCII | | | Caracteres ASCII imprimibles | | | | | | | | |
|-----------------------------|-----|---------------------------|------------------------------|-----|---------|-----|-----|---------|-----|-----|---------|
| DEC | HEX | Símbolo ASCII | DEC | HEX | Símbolo | DEC | HEX | Símbolo | DEC | HEX | Símbolo |
| 00 | 00h | NULL (carácter nulo) | 32 | 20h | espacio | 64 | 40h | @ | 96 | 60h | ` |
| 01 | 01h | SOH (inicio encabezado) | 33 | 21h | ! | 65 | 41h | A | 97 | 61h | a |
| 02 | 02h | STX (inicio texto) | 34 | 22h | " | 66 | 42h | B | 98 | 62h | b |
| 03 | 03h | ETX (fin de texto) | 35 | 23h | # | 67 | 43h | C | 99 | 63h | c |
| 04 | 04h | EOT (fin transmisión) | 36 | 24h | \$ | 68 | 44h | D | 100 | 64h | d |
| 05 | 05h | ENQ (enquiry) | 37 | 25h | % | 69 | 45h | E | 101 | 65h | e |
| 06 | 06h | ACK (acknowledgement) | 38 | 26h | & | 70 | 46h | F | 102 | 66h | f |
| 07 | 07h | BEL (timbre) | 39 | 27h | ' | 71 | 47h | G | 103 | 67h | g |
| 08 | 08h | BS (retroceso) | 40 | 28h | (| 72 | 48h | H | 104 | 68h | h |
| 09 | 09h | HT (tab horizontal) | 41 | 29h |) | 73 | 49h | I | 105 | 69h | i |
| 10 | 0Ah | LF (salto de línea) | 42 | 2Ah | * | 74 | 4Ah | J | 106 | 6Ah | j |
| 11 | 0Bh | VT (tab vertical) | 43 | 2Bh | + | 75 | 4Bh | K | 107 | 6Bh | k |
| 12 | 0Ch | FF (form feed) | 44 | 2Ch | , | 76 | 4Ch | L | 108 | 6Ch | l |
| 13 | 0Dh | CR (retorno de carro) | 45 | 2Dh | - | 77 | 4Dh | M | 109 | 6Dh | m |
| 14 | 0Eh | SO (shift Out) | 46 | 2Eh | . | 78 | 4Eh | N | 110 | 6Eh | n |
| 15 | 0Fh | SI (shift In) | 47 | 2Fh | / | 79 | 4Fh | O | 111 | 6Fh | o |
| 16 | 10h | DLE (data link escape) | 48 | 30h | 0 | 80 | 50h | P | 112 | 70h | p |
| 17 | 11h | DC1 (device control 1) | 49 | 31h | 1 | 81 | 51h | Q | 113 | 71h | q |
| 18 | 12h | DC2 (device control 2) | 50 | 32h | 2 | 82 | 52h | R | 114 | 72h | r |
| 19 | 13h | DC3 (device control 3) | 51 | 33h | 3 | 83 | 53h | S | 115 | 73h | s |
| 20 | 14h | DC4 (device control 4) | 52 | 34h | 4 | 84 | 54h | T | 116 | 74h | t |
| 21 | 15h | NAK (negative acknowle.) | 53 | 35h | 5 | 85 | 55h | U | 117 | 75h | u |
| 22 | 16h | SYN (synchronous idle) | 54 | 36h | 6 | 86 | 56h | V | 118 | 76h | v |
| 23 | 17h | ETB (end of trans. block) | 55 | 37h | 7 | 87 | 57h | W | 119 | 77h | w |
| 24 | 18h | CAN (cancel) | 56 | 38h | 8 | 88 | 58h | X | 120 | 78h | x |
| 25 | 19h | EM (end of medium) | 57 | 39h | 9 | 89 | 59h | Y | 121 | 79h | y |
| 26 | 1Ah | SUB (substitute) | 58 | 3Ah | : | 90 | 5Ah | Z | 122 | 7Ah | z |
| 27 | 1Bh | ESC (escape) | 59 | 3Bh | ; | 91 | 5Bh | [| 123 | 7Bh | { |
| 28 | 1Ch | FS (file separator) | 60 | 3Ch | < | 92 | 5Ch | \ | 124 | 7Ch | |
| 29 | 1Dh | GS (group separator) | 61 | 3Dh | = | 93 | 5Dh |] | 125 | 7Dh | } |
| 30 | 1Eh | RS (record separator) | 62 | 3Eh | > | 94 | 5Eh | ^ | 126 | 7Eh | ~ |
| 31 | 1Fh | US (unit separator) | 63 | 3Fh | ? | 95 | 5Fh | _ | | | |
| 127 | 20h | DEL (delete) | | | | | | | | | |



Actividad: codificación

- **Objetivo:** Aprender como se representa la información en los ordenadores a través de la codificación binaria.
- **Tareas:** Ejercicios de conversión de números de decimal a binario y viceversa y de decimal a hexadecimal y viceversa.
- **Herramientas:**
 - Calculadora para comprobar la conversión:
 - <https://masterplc.com/calculadora/convertir-binario-a-decimal>
 - Completa (desplegables para cambiar las bases a las que se convierte) pero tiene publicidad.



Actividad: codificación

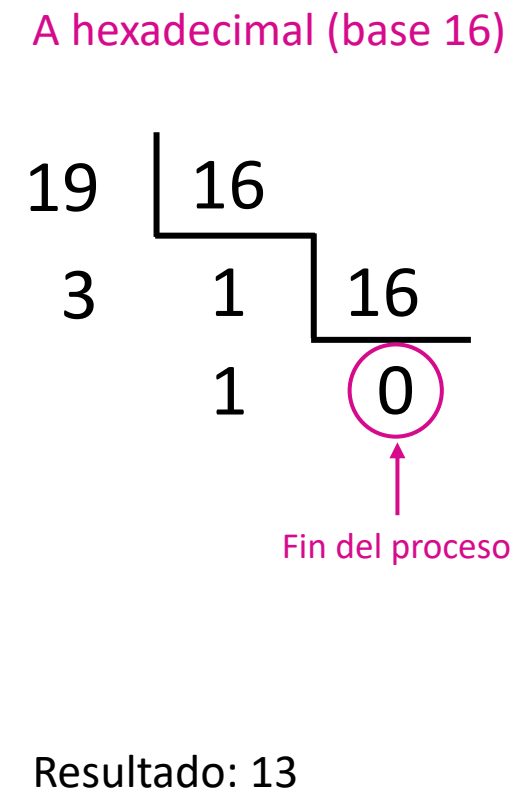
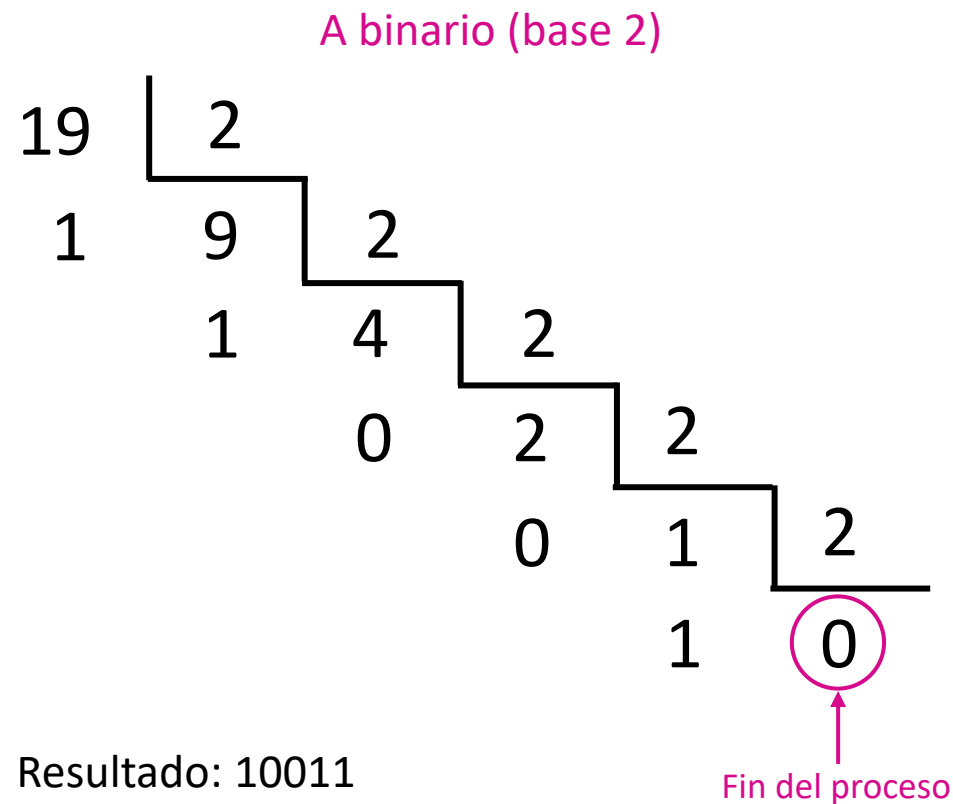
Conversión de decimal (base 10) a otra base

- Se divide número entre la nueva base pasando el resto a ser la primera cifra del nuevo número.
 - En el sistema binario, la base es 2
 - En el sistema hexadecimal, la base es 16
- El cociente resultante se vuelve a dividir entre la base y el resto pasa a ser la siguiente cifra del nuevo número.
- Este proceso se repite hasta que el cociente sea 0.



Actividad: codificación

- Ejemplo de conversión del número decimal 19:





Actividad: codificación

- Ejemplo de conversión de hexadecimal y binario a decimal

De binario (base 2)

Número binario : 10011

$$(10011)_2 = (1 \times 2^0) + (1 \times 2^1) + (0 \times 2^2) + (0 \times 2^3) + (1 \times 2^4) = 19_{10}$$

Número decimal: 19

Cálculo:

$$d_{n-1} \times 2^{n-1} + \dots + d_2 \times 2^2 + d_1 \times 2^1 + d_0 \times 2^0$$

De hexadecimal (base 16)

Número hexadecimal: 13

$$(13)_{16} = (1 \times 16^1) + (3 \times 16^0) = 19_{10}$$

Número decimal: 19

Cálculo:

$$d_{n-1} \times 16^{n-1} + \dots + d_2 \times 16^2 + d_1 \times 16^1 + d_0 \times 16^0$$

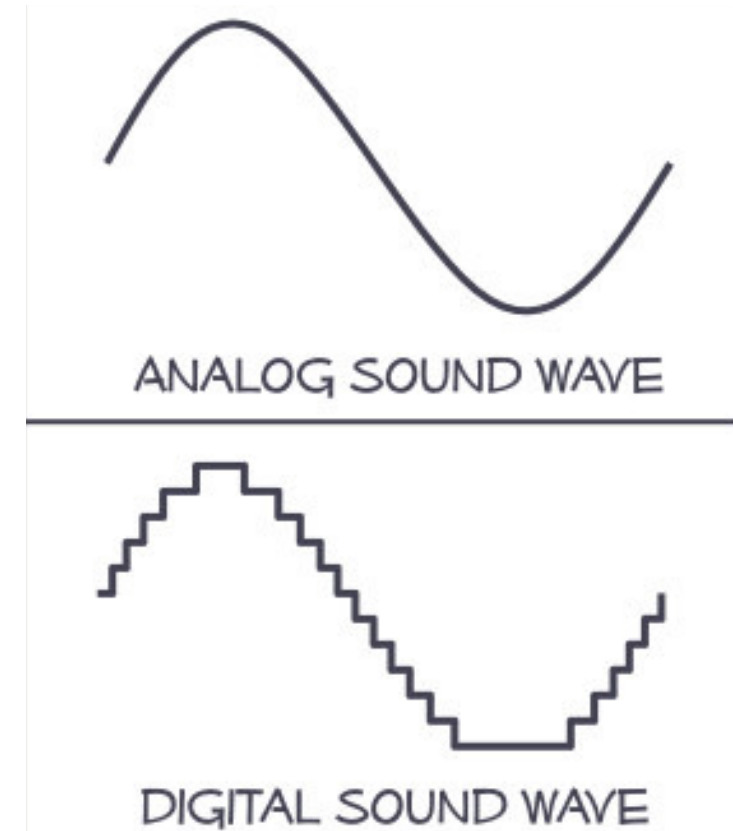
Los dígitos se numeran de derecha a izquierda. En el número hexadecimal 13: $d_0 = 3$, $d_1 = 1$



Representación de la percepción

Sonido o audio digital:

- Las ondas sonoras son analógicas.
- Almacenar el audio digital requiere su conversión a un formato digital.
- Este proceso se conoce como **digitalización**.



Fuente: <https://www.linkedin.com/pulse/what-digital-analog-audio-eilbron-khoshabeh/>

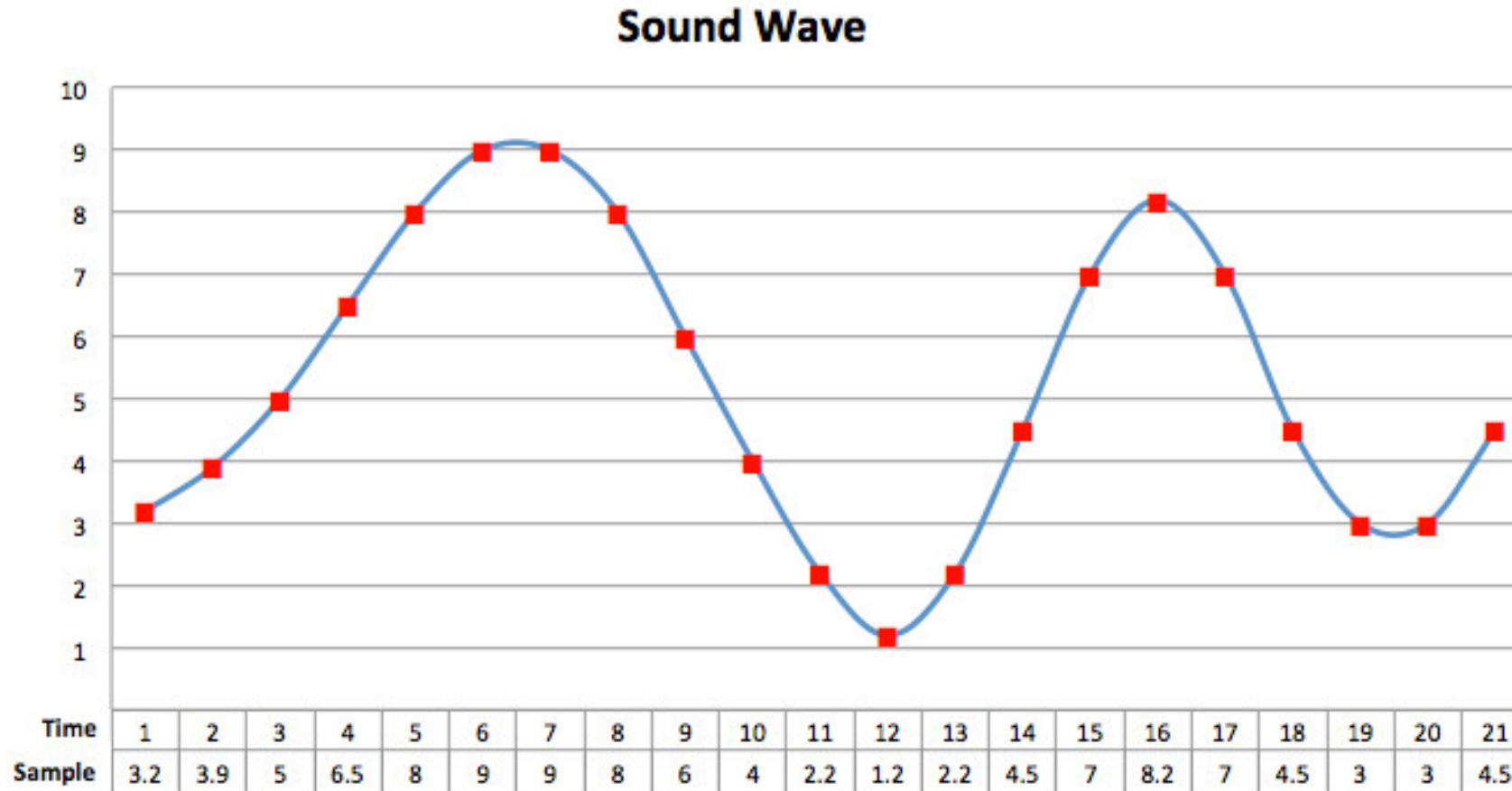


Representación de la percepción

Factores que influyen en la digitalización:

- **Frecuencia de muestreo:** Constituye el **número de muestras registradas cada segundo** y cuanto mayor sea, mejor será la calidad del sonido (más muestras por segundo).
- **Tasa de bits:** Expresa el número de **bits utilizados para almacenar** cada muestra y cuantos más bits se utilicen, mayor será la precisión de la representación, pero también aumentará el tamaño del archivo de audio final.

Representación de la percepción

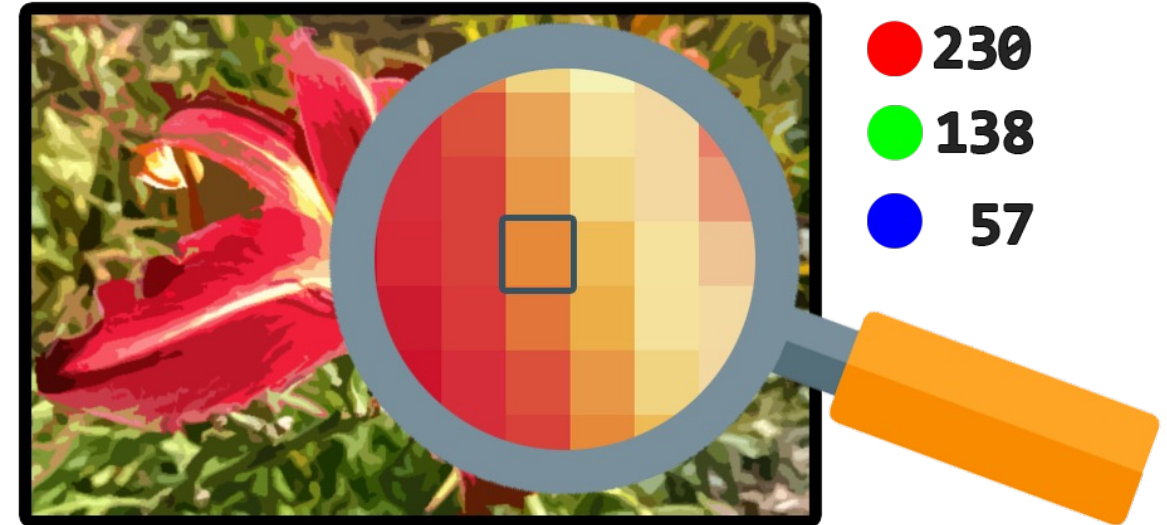


(c) teachwithict.weebly.com

Representación de la percepción

Imágenes digitales:

- Se representan como una matriz, donde cada elemento de la matriz (llamado píxel) tiene un valor que representa un color.
- El color se representa con modelos de color, uno de los más comunes es RGB (rojo, verde, azul).

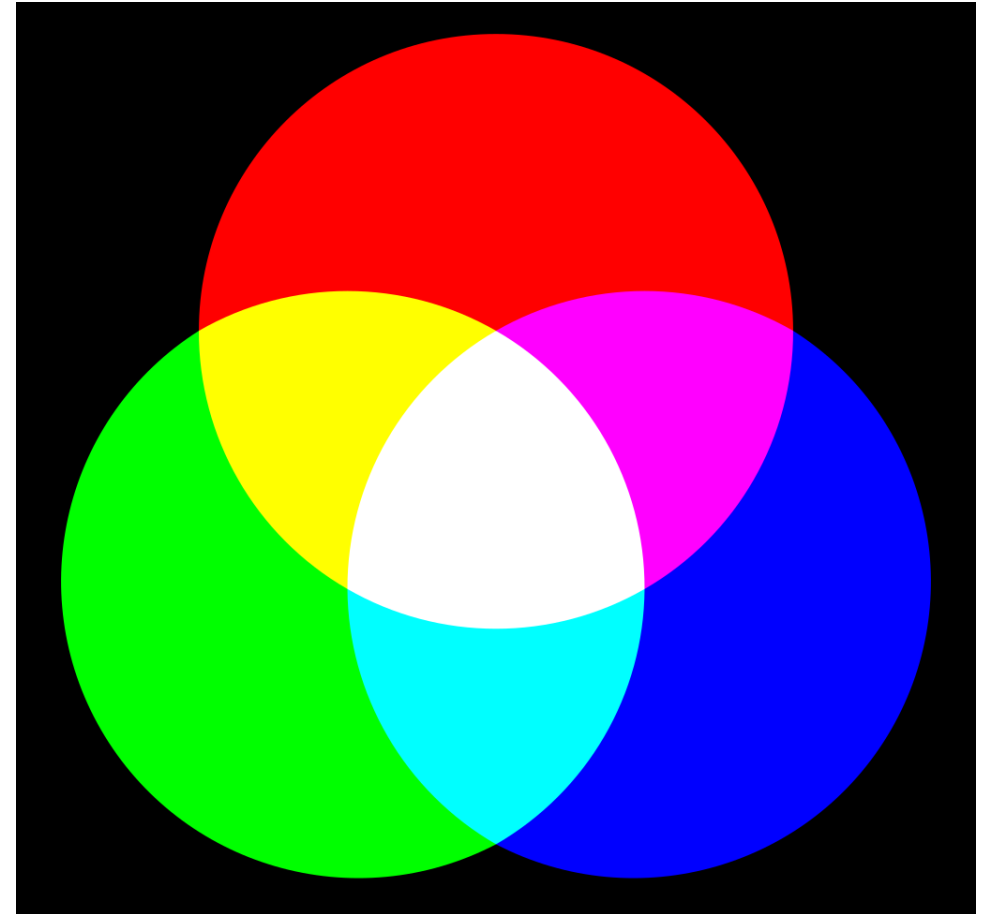




Representación de la percepción

Modelo de color RGB

- Modelo aditivo
- El color se obtiene por mezcla de 3 colores primarios.





Actividad: imágenes

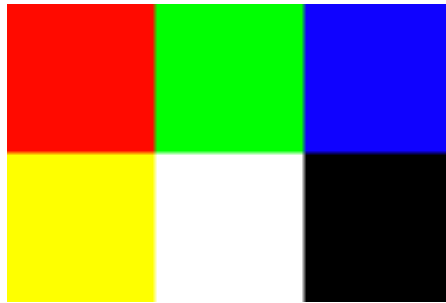
- **Objetivo:** Comprender la representación interna de las imágenes.
- **¿Cómo?:** Utilizaremos un formato de imágenes (PPM) que permite ver el contenido del archivo de imagen como un archivo de texto:
 1. Abriremos el archivo de la imagen desde un editor de texto como el bloc de notas y también desde un programa de visualización de imágenes.
 2. Modificaremos el contenido de la imagen en el editor de texto y veremos el resultado en el programa de visualización de imágenes.



Actividad: imágenes

- Utilizaremos estas dos imágenes imagen de 6 píxeles y 9 píxeles:

imagen_6px.ppm



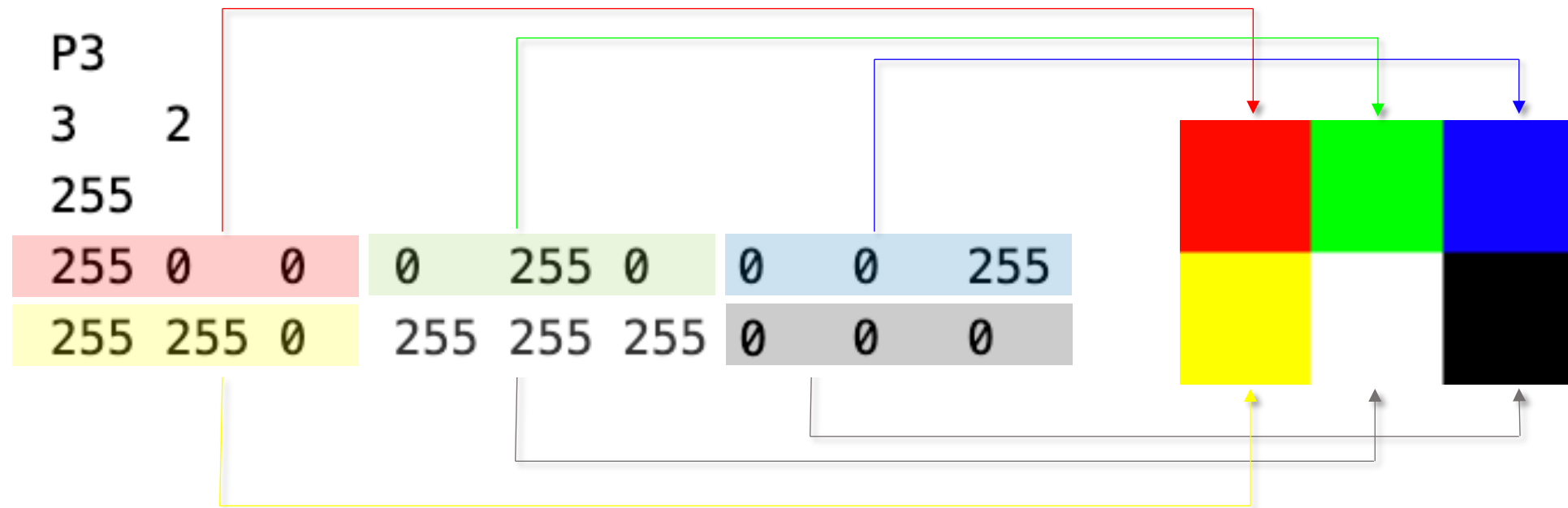
cruz_9px.ppm



- Para visualizar una imagen en formato PPM podemos utilizar:
 - Windows: <https://www.irfanview.com/>
 - MacOS: Preview (está instalado por defecto)
 - Linux: <https://www.gimp.org/>

Actividad: imágenes

- Ejemplo del contenido de la imagen_6px.ppm



Representación de la percepción

- **Texto**

- Token

Hola, ¿cómo estás?

Palabras

"Hola"
" "
,
"¿"
"cómo"
"estás"
"?"

Caracteres

"H"
"o"
"l"
"a"
", "
,
...

Longitud fija

"Ho"
"la"
",¿"
"co"
"mo"
...

Longitud variable

"Hola"
" "
,
"¿"
"cómo"
"estás"
"?"



Representación del conocimiento

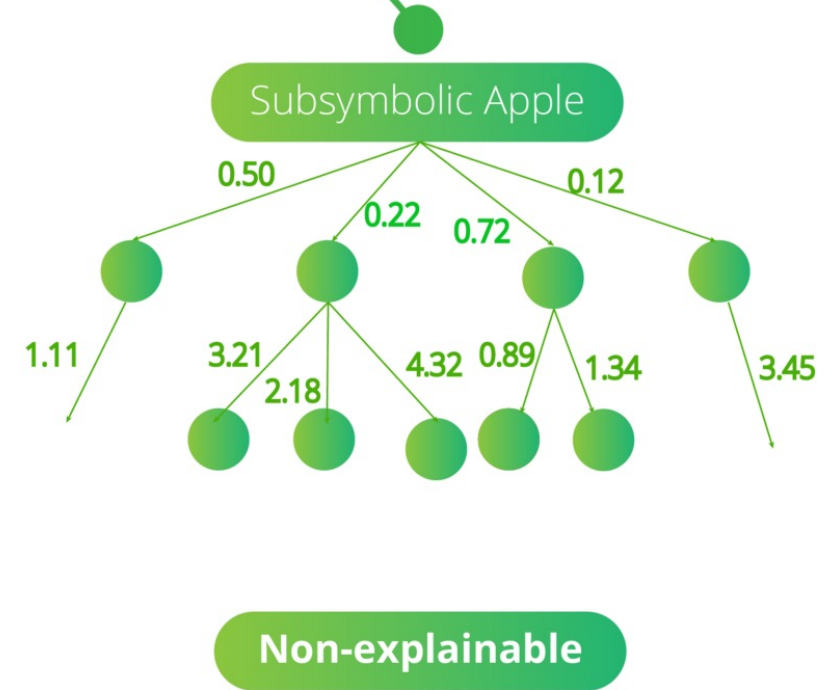
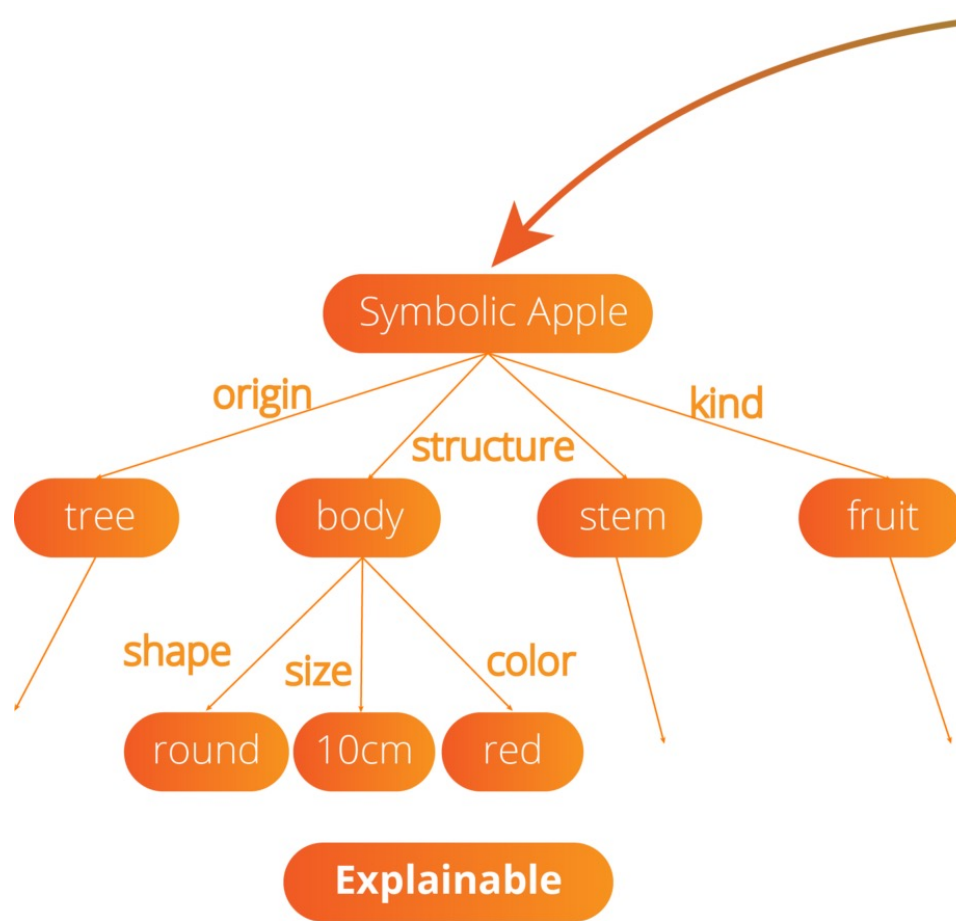
IA Simbólica

- Representación: símbolos (unidades discretas)
- Se pueden combinar en estructuras más complejas: árboles de decisión, reglas, relaciones, grafos, etc.

IA Subsimbólica

- Representaciones de la información menos explícitas. Ej: redes de neuronas.
- Aprenden directamente de los datos, identificando patrones.

Representación del conocimiento





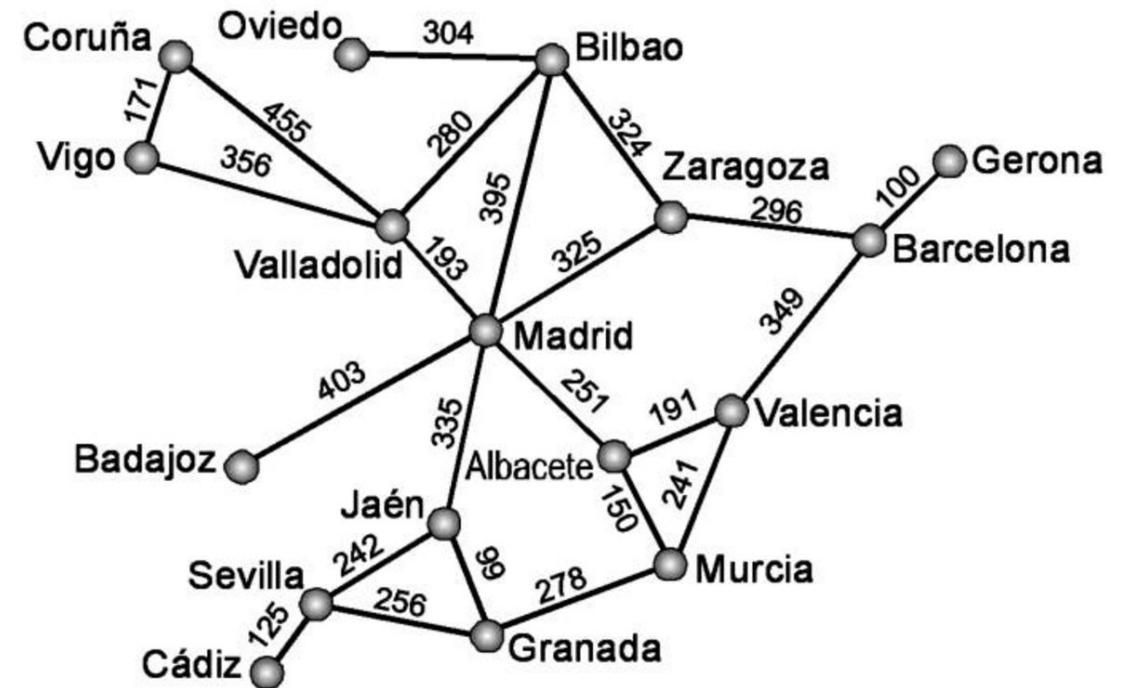
Representación del conocimiento

GRAFOS

- Estructuras matemáticas conformadas por un conjunto de **nod**os (vértices) y un conjunto de **aristas** (conexiones) que establecen relaciones entre los nodos.
- Los grafos pueden ser **dirigidos** si existe una dirección específica en las relaciones que representan y, por tanto, indican cómo deben ser recorridos, o bien **no dirigidos**.

Representación del conocimiento

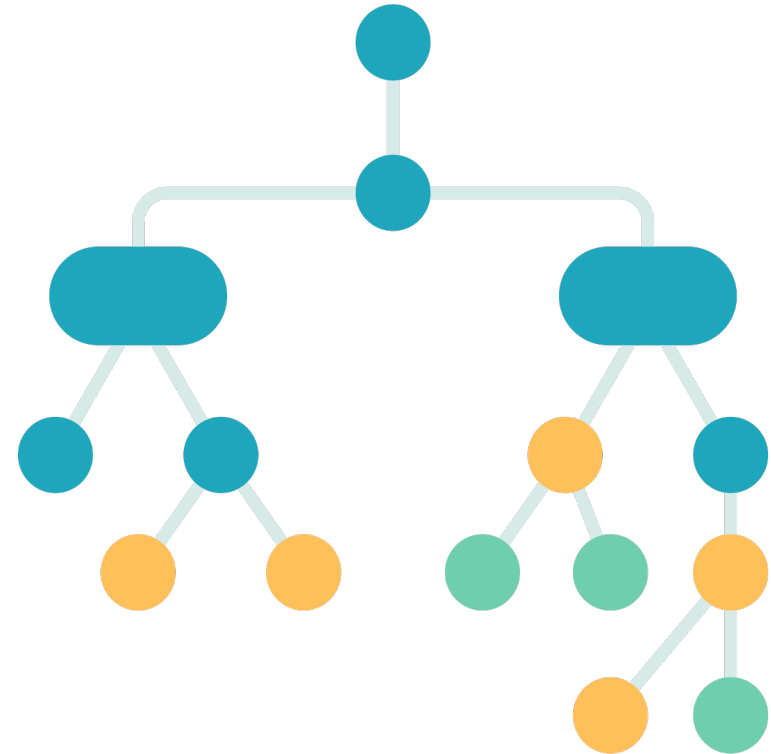
- Grafo no dirigido
- Representa conexiones entre ciudades (nodos) y su distancia asociada (aristas)



Representación del conocimiento

ÁRBOLES

- Estructura de datos jerárquica que consta de nodos conectados por aristas dirigidas.
 - El nodo superior o inicial se denomina raíz.
- Como en los grafos, cada nodo puede representar una entidad, objeto o concepto, mientras que las aristas representan las relaciones que se establecen entre ellos.





Representación del conocimiento

TABLAS

- Se utilizan para almacenar datos organizados, normalmente:
 - Cada columna representa un campo o atributo.
 - Cada fila representa un registro.
- Ejemplo: tabla de estudiantes con nombre, apellidos, edad; datos de salud, ...

| Age | Sex | BP | Cholesterol |
|-----|-----|--------|-------------|
| 23 | F | HIGH | HIGH |
| 47 | M | LOW | HIGH |
| 47 | M | LOW | HIGH |
| 28 | F | NORMAL | HIGH |
| 61 | F | LOW | HIGH |
| 22 | F | NORMAL | HIGH |
| 49 | F | NORMAL | HIGH |
| 41 | M | LOW | HIGH |
| 60 | M | NORMAL | HIGH |
| 43 | M | LOW | NORMAL |



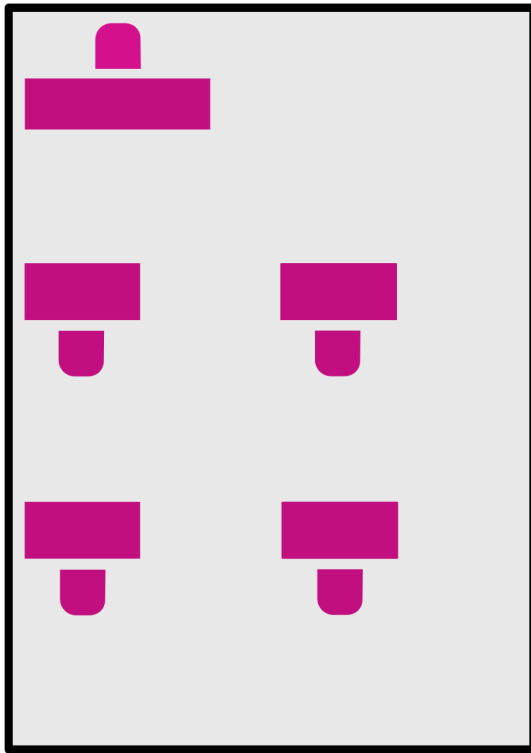
Representación del conocimiento

TABLAS

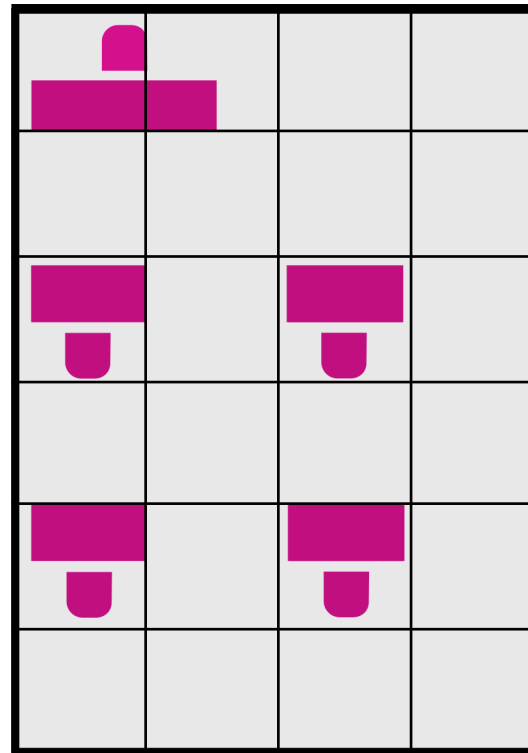
- Puede utilizarse también para representar un mapa del entorno en el que se encuentra el agente.
- Este tipo de mapas se conocen como cuadrícula de ocupación.
- El objetivo es convertir un entorno con posiciones infinitas en una cuadrícula finita.
 - Variando el número de filas y columnas de la cuadrícula, variaremos la precisión.

Representación del conocimiento

Cuadrícula de ocupación



Entorno



Entorno discretizado

| | | | |
|---|---|---|---|
| 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 |

Cuadrícula de ocupación



Representación del conocimiento

Cuadrícula de ocupación

- Sobre ella se pueden aplicar algoritmos de búsqueda de la mejor ruta entre dos puntos.
- En el vídeo se explica:
 - Wavefront (cuadrícula)
 - Dijkstra (grafo)



Bot Field. Wavefront And Dijkstra | Path Planning
2020. YouTube (<https://youtu.be/GMC4M14gTOA>)



Representación del conocimiento

- **SLAM:** Localización y mapeado simultáneo.
- Ejemplo en un entorno dinámico.



FZlchannel. Follow-Me AGV based on SLAM and Dynamic Navigation in Large-Scale Environments 2021. YouTube (<https://youtu.be/Gac5UBkLHAK>)



Razonamiento

- **Resolución de problemas**
 - Algoritmos
- **Programas**
 - Conjunto de instrucciones que se ejecutan en un ordenador
 - Implementan un determinado algoritmo

LENGUAJE DE ALTO NIVEL
(JAVA, C#, PHP, PYTHON, ETC)

LENGUAJE DE BAJO NIVEL
ENSAMBLADOR

CÓDIGO MÁQUINA
(BINARIO [0-1])

HARDWARE
(PARTE FISICA DE LA COMPUTADORA)

Lenguaje alto nivel (Python)

```
suma = suma + número
```

Ensamblador

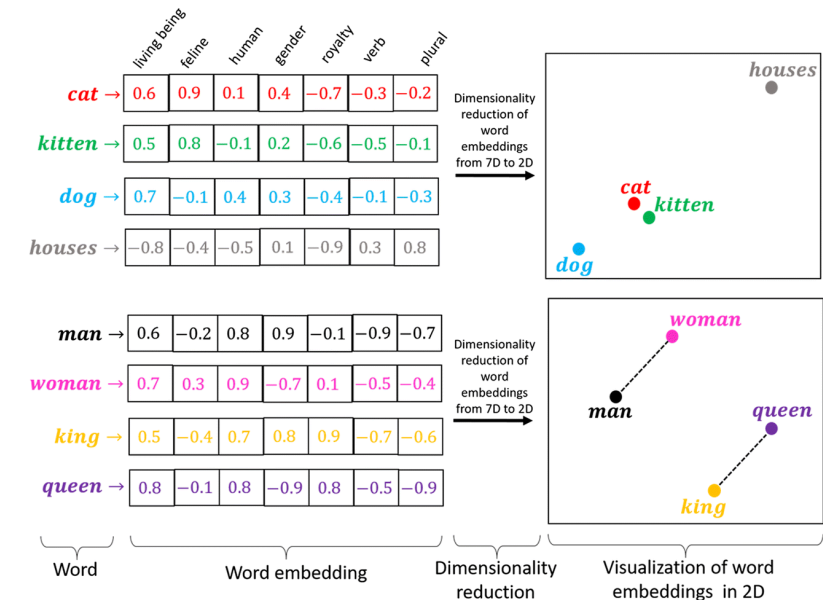
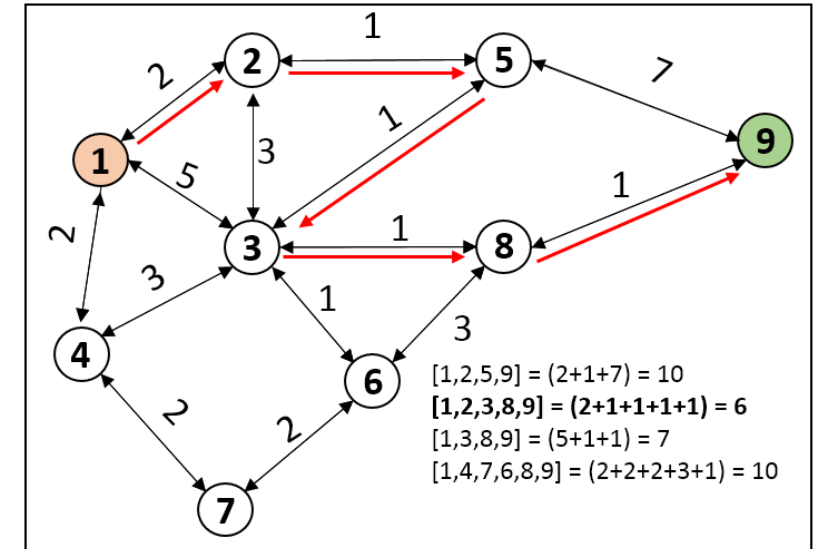
```
mov ax, [suma]  
add ax, [numero]  
mov [suma], ax
```

Código máquina

```
A19401  
03063204  
A39401
```

Programas en IA

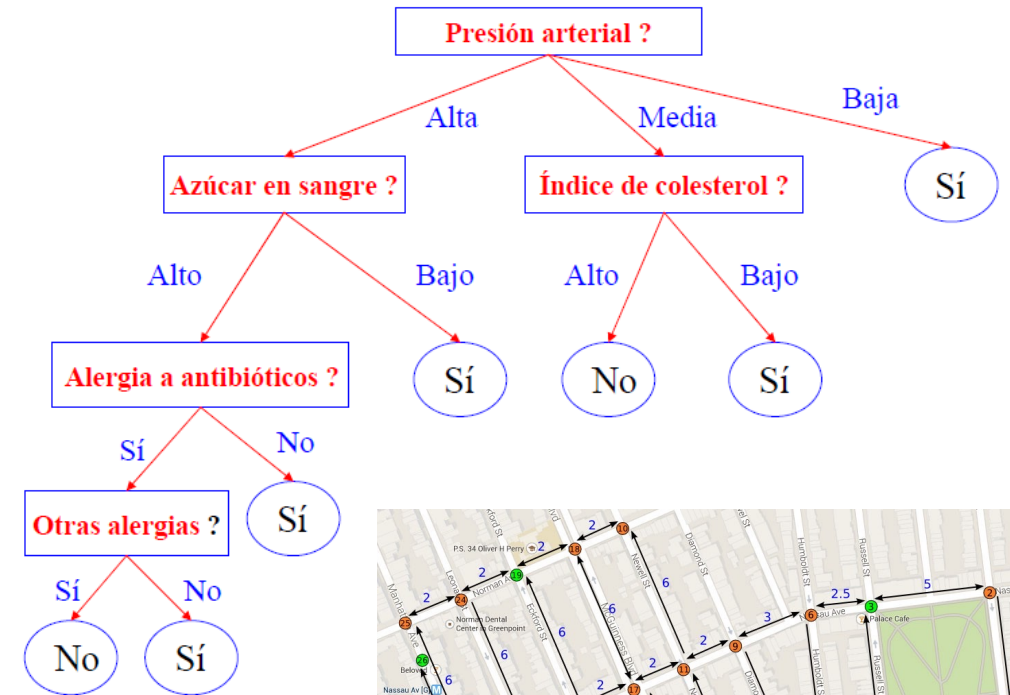
- Un programa sencillo opera sobre una representación simple:
 - suma = $a + b$
- Un programa que tenga que calcular la ruta óptima de un coche requiere una representación del mapa de carreteras, y el algoritmo es complejo
- Un programa que mantenga una conversación con un humano requiere una representación más abstracta que incluye el lenguaje



Representación y razonamiento

- Acoplamiento entre representación y razonamiento
- Representación simbólica:
 - Razonamiento lógico utilizando técnicas lógicas como la deducción y la inducción.
- Representación numérica:
 - Estrategias de búsqueda y optimización de funciones

Ejemplo: Árbol de decisión para “¿ Administrar fármaco F ?”



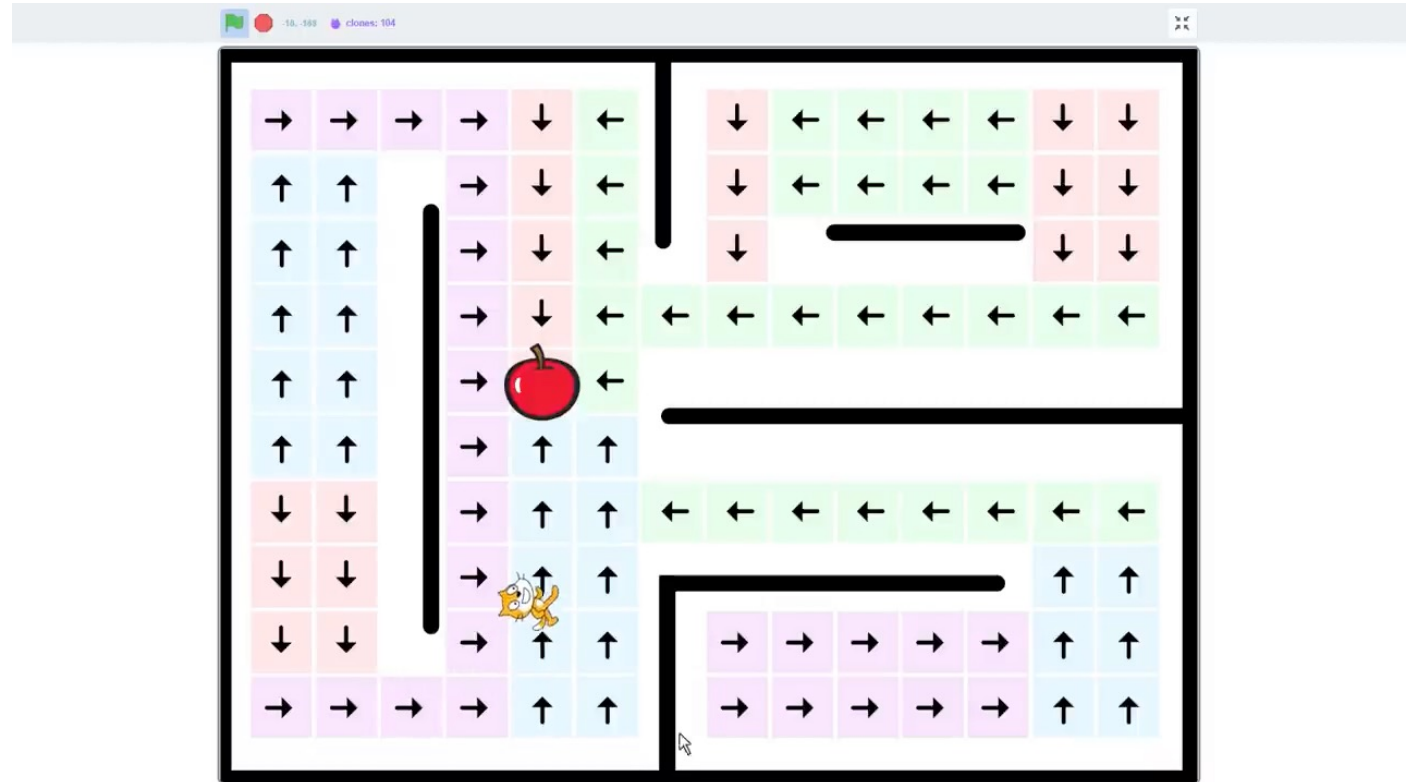


Actividad: Seguir la ruta

Objetivo: Buscar el camino a seguir desde una posición inicial hasta un objetivo.

¿Cómo?: Utilizando un programa en Scratch, en el que se representará un mapa, se fijará un objetivo se crearán las rutas y se enseñará al agente (gato) a seguir la ruta hasta llegar al objetivo.

- El vídeo contiene un tutorial paso a paso del desarrollo del programa.



Simple Pathfinding Tutorial – grifpatch
2021. YouTube (<https://youtu.be/OfaPPgOT--E>)



Actividad: Akinator

- **Objetivo 1:** entender como los árboles de decisión pueden representar los procesos de razonamiento y toma de decisiones.
- **Objetivo 2:** Aprender como se puede utilizar ingeniería inversa para desentrañar el conocimiento implícito de un sistema automatizado.



<https://es.akinator.com/>



Actividad: Akinator

Desarrollo de la actividad:

- **Jugar con Akinator:** El alumnado juega de forma individual a Akinator (<https://es.akinator.com/>) durante un tiempo determinado y toma notas detalladas de varias partidas. Se espera que identifiquen las preguntas clave que los llevó al final del juego y como se relacionan esas preguntas entre sí.
- **Diseño de árboles individuales:** Cada alumno creará un árbol de decisión basado en sus notas. Deben representar las preguntas como nodos, las posibles respuestas como ramas y los personajes adivinados como las hojas del árbol.
- **Puesta en común y creación de árbol global:** En pequeños grupos, colaboran para combinar sus árboles individuales en un árbol de decisión único, más general, que represente mejor el proceso de razonamiento de Akinator.
- **Exposición** del árbol de decisión de cada grupo al resto de los grupos del aula.



Actividad: Adivina el animal

Objetivo del juego: adivinar el animal en el que está pensando una persona haciendo una serie de preguntas de sí o no.

¿Cómo?: Con un árbol de decisión:

1. Dibujar el árbol de decisión para poder realizar el proceso de razonamiento.
2. El procedimiento para jugar es fijo: se comienza con una pregunta y al llegar a cualquier nodo no terminal, se debe hacer la pregunta del nodo, y luego se sigue la rama "sí" o "no" para llegar al siguiente nodo.
3. Al llegar a un nodo terminal, se indica el animal asociado con ese nodo y espera para ver si la suposición es correcta.



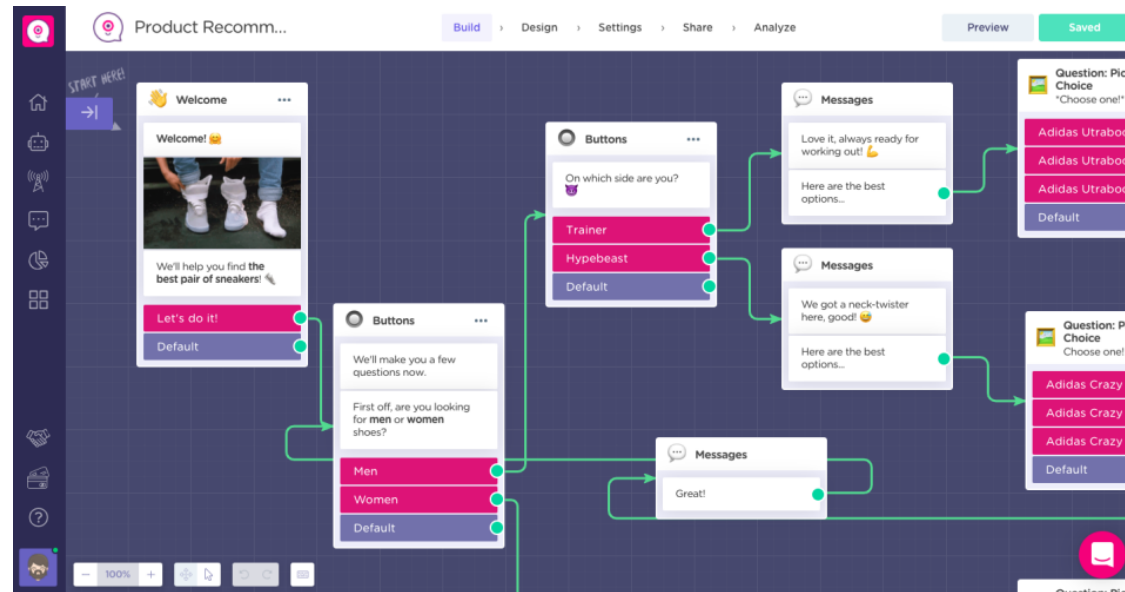
Adivina el animal





Chatbot

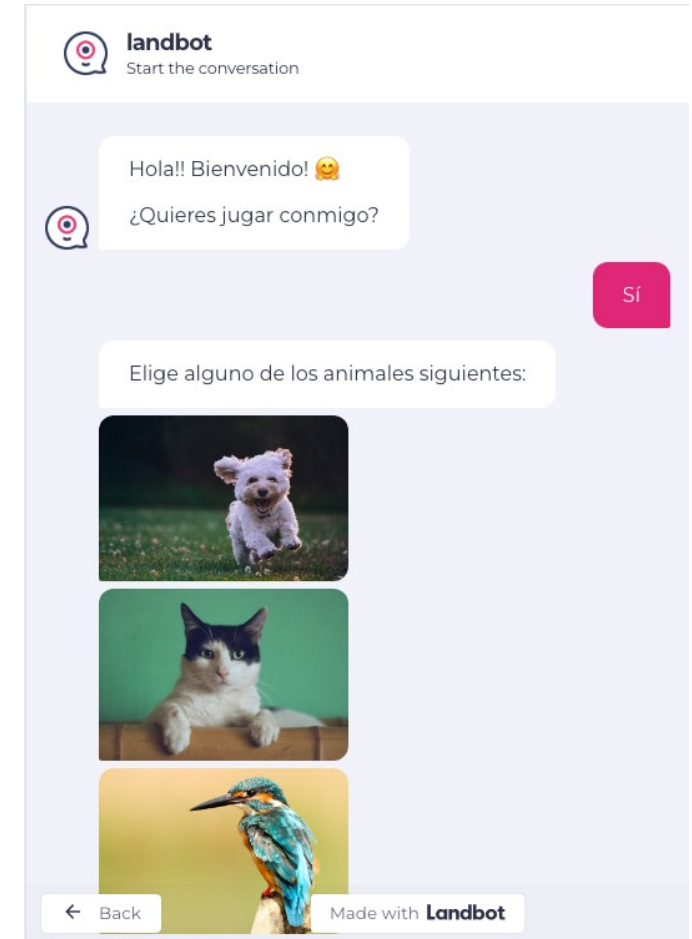
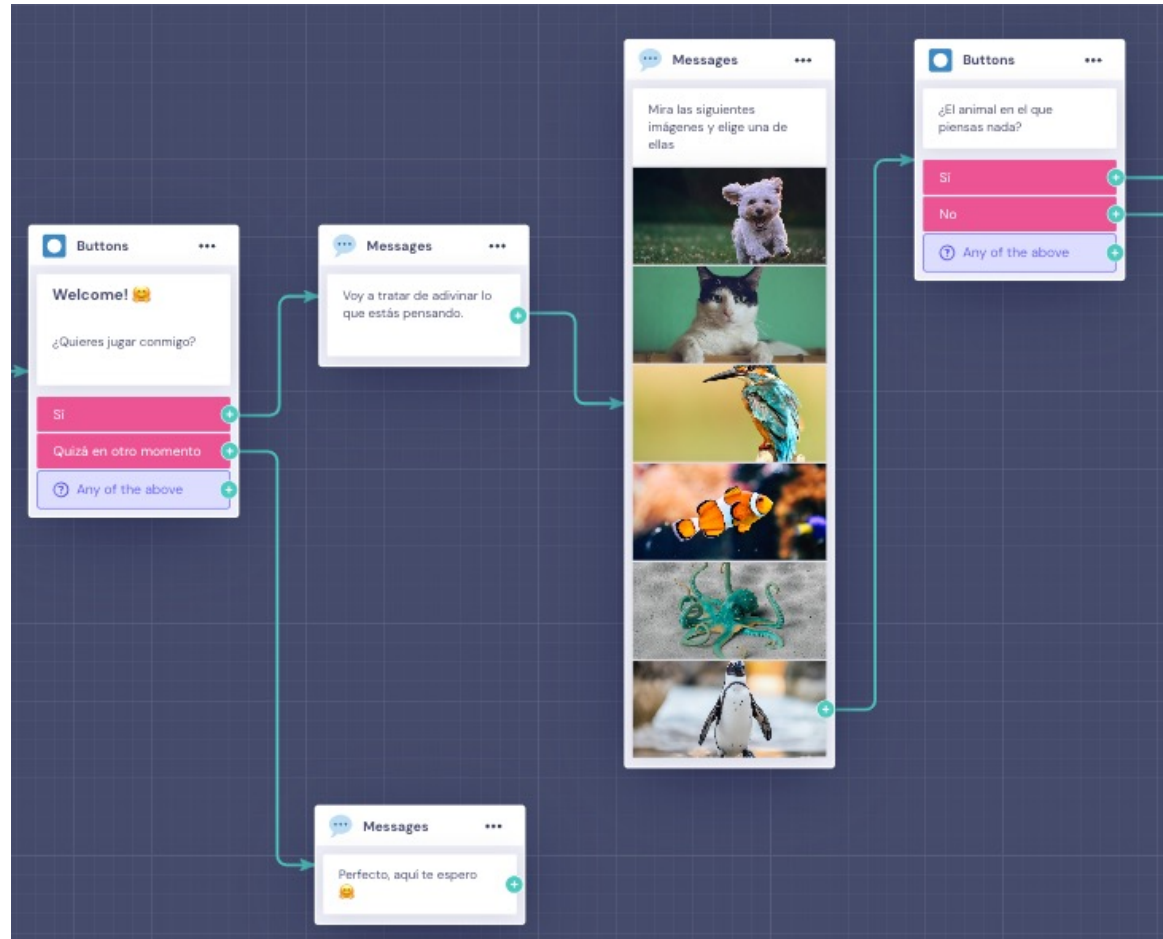
- **Creación de un chatbot:**
- **Landbot.io** es una solución basada na nube para crear sitios web conversacionales para interactuar con los usuarios.
- Este recurso está pensado para pequeñas empresas.





Chatbot

- CHATBOT



Ejemplo de chatbot: <https://landbot.online/v3/H-2659135-VUWAKK8LIE2B34AS/index.html>



Scratch



Video disponible en: <https://www.edu.xunta.gal/centros/cfrferrol/aulavirtual/course/view.php?id=1755#section-3>



Scratch

- Pasos a seguir:
 - Decir al usuario que piense en uno de los animales y que pulse la tecla espacio.
 - Ir haciendo preguntas y permitir al usuario que responda "Sí" o "No".
- Desarrollo por fases:
 - Versión 1:
 - Cada vez que se juega se adivina una palabra y se termina. Ejemplo en la diapositiva siguiente.
 - Versión 2:
 - Que el juego se repita hasta que el usuario diga que no quiere seguir jugando.
 - En esta versión dividiremos el programa en funciones.



Scratch

```
say Piensa en alguno de los siguientes animales, cuando hayas acabado pulsa la tecla espacio: perro, pingüino, pez, pájaro, pulpo o gato.
wait until key space pressed?
ask ¿El animal en el que piensas nada? Responde "Si" o "No" and wait
if answer = Si then
  ask ¿Tiene escamas? and wait
  if answer = Si then
    think Hmm... for 2 seconds
    say Es un pez!
  else
    ask ¿Tiene tentáculos? and wait
    if answer = Si then
      think Hmm... for 2 seconds
      say Es un pulpo!
    else
      think Hmm... for 2 seconds
      say Es un pingüino!
```

```
else
  ask ¿Vuela? and wait
  if answer = Si then
    think Hmm... for 2 seconds
    say Es un pájaro!
  else
    ask ¿Ladra? and wait
    if answer = Si then
      think Hmm... for 2 seconds
      say Es un perro!
    else
      think Hmm... for 2 seconds
      say Es un gato!
```



XUNTA
DE GALICIA

CENTRO DE
FORMACIÓN E
RECURSOS DE FERROL

Inteligencia Artificial para la Sociedad



Alma Mallo

alma.mallo@udc.es

Noviembre de 2024