

# Introducción a la Inteligencia Artificial para educación pre-universitaria

---



Francisco Bellas

CFR Ferrol

Octubre 2021



# Iniciación a la IA (1)

Obxectivos	Contidos	Criterios de avaliación	Estándares de aprendizaxe	Competencias clave
Bloque 1. Que é a Inteligencia Artificial				
<ul style="list-style-type: none"><li>a</li><li>b</li><li>e</li><li>h</li><li>i</li></ul>	<ul style="list-style-type: none"><li>B1.1 Que é a Inteligencia Artificial?<ul style="list-style-type: none"><li>B1.2 Inteligencia natural fronte a Inteligencia Artificial</li><li>B1.3 Historia da Inteligencia Artificial</li><li>B1.4 Inteligencia Artificial forte fronte a Inteligencia Artificial feble</li></ul></li></ul>	<ul style="list-style-type: none"><li>B1.1 Coñecer á orixe da IA, a que campo de coñecemento pertence, a súa vinculación coa intelixencia humana e animal, e os dous principais enfoques da mesma.</li></ul>	<ul style="list-style-type: none"><li>IIAB1.1.1 Define o significado de Inteligencia Artificial e sabe diferenciala da intelixencia natural.</li><li>IIAB1.1.2 Identifica o campo da Inteligencia Artificial dentro do campo de coñecemento adecuado (ciencias da computación)</li><li>IIAB1.1.3 Coñece a diferenza entre a Inteligencia Artificial forte e feble.</li></ul>	<ul style="list-style-type: none"><li>CCL</li></ul>
<ul style="list-style-type: none"><li>d</li><li>i</li><li>n</li></ul>	<ul style="list-style-type: none"><li>B1.5 Elementos dun sistema intelixente<ul style="list-style-type: none"><li>B1.6 Contornas reais, simuladas e virtuais</li><li>B1.7 Bloques básicos dun sistema de IA (percepción, representación, razoamento, aprendizaxe e actuación)</li></ul></li></ul>	<ul style="list-style-type: none"><li>B1.2 Coñecer os compoñentes básicos dun sistema de IA, entendendo que está situado nunha contorna real ou virtual coa que interactúa, e que a complexidade dos diferentes bloques pode variar.</li></ul>	<ul style="list-style-type: none"><li>IIAB1.2.1 Identifica os elementos básicos dun sistema intelixente.</li><li>IIAB1.2.2 Distingue e define os diferentes tipos de contornas nos que pode estar situado un sistema intelixente</li></ul>	<ul style="list-style-type: none"><li>CAA</li><li>CSIEE</li></ul>
<ul style="list-style-type: none"><li>a</li><li>d</li><li>i</li></ul>	<ul style="list-style-type: none"><li>B1.8 Campos de aplicación da Inteligencia Artificial</li></ul>	<ul style="list-style-type: none"><li>B1.3 Coñecer os principais campos de aplicación real da IA (IA médica, robótica intelixente, contornas intelixentes: smart building, smart city, smart factory; sistemas de recomendación, videoxogos, chatbots, etc) e identificar os bloques básicos dun sistema intelixente en casos de uso concretos.</li></ul>	<ul style="list-style-type: none"><li>IIAB1.3.1 Recoñece cando un sistema aplicado está baseado en IA ou non</li><li>IIAB1.3.2 Identifica os bloques básicos dun sistema intelixente en exemplos concretos de sistemas de IA en funcionamento</li></ul>	<ul style="list-style-type: none"><li>CD</li><li>CAA</li><li>CCEC</li></ul>
Bloque 2. Áreas básicas da IA				
<ul style="list-style-type: none"><li>c</li><li>f</li><li>g</li></ul>	<ul style="list-style-type: none"><li>B2.1 Percepción e actuación en IA<ul style="list-style-type: none"><li>B2.2 Sensorización contra percepción</li><li>B2.3 Sensores e actuadores básicos (distancia, orientación, luz, cor, motores, rodas, brazos)</li><li>B2.4 Sensores e percepción no ámbito da IA (cámaras e visión artificial, micrófonos e recoñecemento da fala, pantallas e interacción táctil)</li><li>B2.5 Actuadores e accións no ámbito da IA (altofalantes e produción de fala, navegación, manipulación, pantallas e outros interfaces virtuais)</li><li>B2.6 Interacción humano-máquina</li></ul></li></ul>	<ul style="list-style-type: none"><li>B2.1 Distinguir sensorización e percepción, coñecer os sensores e actuadores máis relevante na IA, coñecer a relevancia da interacción humano-máquina. Saber utilizar sensores e actuadores reais no ámbito da IA.</li></ul>	<ul style="list-style-type: none"><li>IIAB2.1.1 Comprende a relevancia dos sensores e actuadores nos sistemas de IA, tanto reais como virtuais.</li><li>IIAB2.1.2 Distingue os sensores e actuadores propios dos sistemas intelixentes e por que proporcionan información de maior complexidade.</li><li>IIAB2.1.3 Coñece a relevancia da interacción humano-máquina na Inteligencia Artificial e comprende que todo sistema intelixente debe estar adaptado ás necesidades do público ao que vai dirixido.</li></ul>	<ul style="list-style-type: none"><li>CD</li><li>CSIEE</li></ul>
<ul style="list-style-type: none"><li>c</li><li>f</li><li>g</li></ul>	<ul style="list-style-type: none"><li>B2.7 Aprendizaxe automática<ul style="list-style-type: none"><li>B2.8 Conceptos básicos: preparación dos datos, aprendizaxe dos modelos e análise dos resultados</li><li>B2.9 Supervisado (clasificación e regresión)</li><li>B2.10 Non supervisado (agrupamento)</li><li>B2.11 Por reforzo (q-learning)</li></ul></li></ul>	<ul style="list-style-type: none"><li>B2.2 Coñecer os fundamentos da aprendizaxe automática, programación baseada nos datos, tratamento dos datos (conxuntos de adestramento e test), tipos de modelos básicos, análise de resultados. Comprender as diferenzas entre os 3 tipos de aprendizaxe. Saber utilizar ferramentas básicas de aprendizaxe de modelos, e lograr un axuste de parámetros apropiado.</li></ul>	<ul style="list-style-type: none"><li>IIAB2.2.1 Coñece que é o aprendizaxe automático e os seus fundamentos</li><li>IIAB2.2.2 Selecciona correctamente os datos para realizar o axuste dun modelo.</li><li>IIAB2.2.3 Utiliza adecuadamente ferramentas de aprendizaxe de modelos e logra analizar os resultados con rigor, comprendendo os factores que influencian o resultado.</li></ul>	<ul style="list-style-type: none"><li>CMCCT</li><li>CSIEE</li></ul>



# Iniciación a la IA (2)

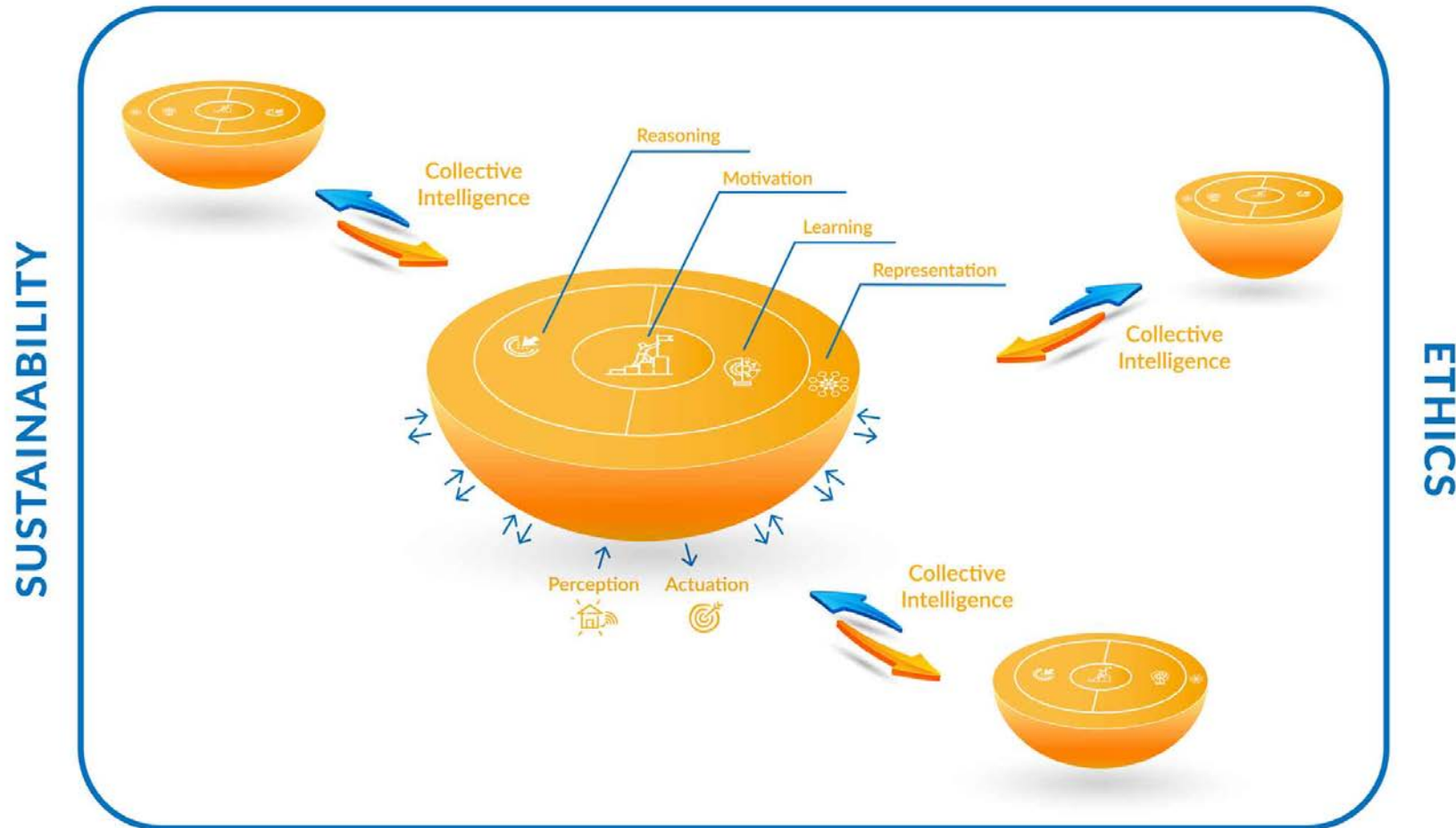
Obxectivos	Contidos	Criterios de avaliación	Estándares de aprendizaxe	Competencias clave
			• IIAB2.2.4 Diferenza os tres tipos de aprendizaxe	
• c • f • g • m	• B2.12 Representación e razoamento <ul style="list-style-type: none"><li>o B2.13 Como representar o coñecemento?</li><li>o B2.14 Grafos e árbores de decisión</li><li>o B2.15 Busca básica</li><li>o B2.16 Fundamentos do razoamento probabilístico</li></ul>	• B2.3 Comprender como se representa computacionalmente o coñecemento a partir das percepcións, e como esta representación pode ser utilizada para os procesos de razoamento. Implementar programas que resolvan problemas sinxelos sobre árbores e grafos, utilizando algoritmos de busca sinxelos. Coñecer os fundamentos do razoamento probabilístico	• IIAB2.3.1 Comprende como se representan computacionalmente os datos e como se utiliza esta representación nos procesos de razoamento.  • IIAB2.3.2 Deseña árbores de decisión e grafos para resolver problemas sinxelos.	• CMCC • CAA
• c • f • g • m	• B2.17 IA colectiva <ul style="list-style-type: none"><li>o B2.18 Comunicación do coñecemento</li><li>o B2.19 Contornas intelixentes</li></ul>	• B2.4 Coñecer as potencialidades da transmisión de información e coñecemento entre sistemas de IA. Comprender os fundamentos da IoT (Internet of Things) como base das contornas intelixentes: casas, edificios, cidades, fábricas.	• IIAB2.4.1 Comprende que os sistemas de IA futuros estarán interconectados formando parte dun ecosistema de IA colectiva (fontes de información e coñecemento distribuídas)	• CD
Bloque 3. Impacto da IA				
• a • b • e • g	• B3.1 Ética da IA	• B3.1 Coñecer as consecuencias sociais do uso da IA en niveis como: a igualdade de raza e xénero, o desemprego, a toma de decisións morais e a influencia e desafío da privacidade que ten sobre os usuarios. Distinguir entre mitos e realidades da IA	• IIAB3.1.1 Identifica as consecuencias sociais do uso da IA e comprende as súas vantaxes e posibles riscos	• CCL • CSC • CCEC
• a • b • c • d • e	• B3.2 Aspectos legais da IA	• B3.2 Coñecer as implicacións legais do uso de sistemas autónomos e intelixentes	• IIAB3.2.1 Comprende as implicacións legais do uso de sistemas intelixentes, e identifica os posibles baleiros legais que existen sobre a IA dada a súa curta existencia.	• CAA • CSC
• a • b • c • p	• B3.3 Sostibilidade	• B3.3 Coñecer as consecuencias do crecemento de sistemas de IA na pegada do carbono, os residuos informáticos, o uso de redes de comunicacións. Coñecer a impacto positivo da IA nos Obxectivos de Desenvolvemento Sostible (ODS).	• IIAB3.3.1 Define o significado de sostibilidade e recoñece as consecuencias que trae o crecemento de sistemas de IA no relativo a este aspecto. Comprende os impactos positivos da IA nos ODS.	• CSC



# Organización del curso

- Sesión 1: Introducción a la IA
- Sesión 2: Percepción y actuación en IA
- **Sesión 3: Representación y razonamiento**
- Sesión 4: Aprendizaje automático
- Sesión 5: IA colectiva
- Sesión 6: Impacto social de la IA

## LEGAL ASPECTS OF AI





# Contenidos sesión 3

- Representación y razonamiento
  - Representación de la información
  - Representación y razonamiento en IA
    - Razonamiento simbólico
    - Árboles de decisión
    - Mapas
    - Grafos
    - Búsqueda



# Representación y razonamiento

- ¿Qué son?
- “Los agentes mantienen representaciones del mundo y las utilizan para el razonamiento”

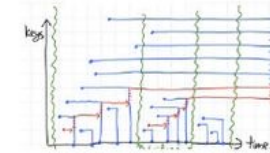




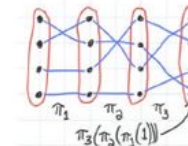


# Representación y razonamiento

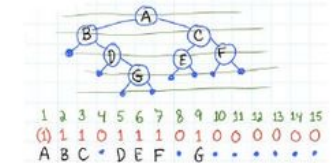
- Pensamiento computacional
- Resolución de problemas



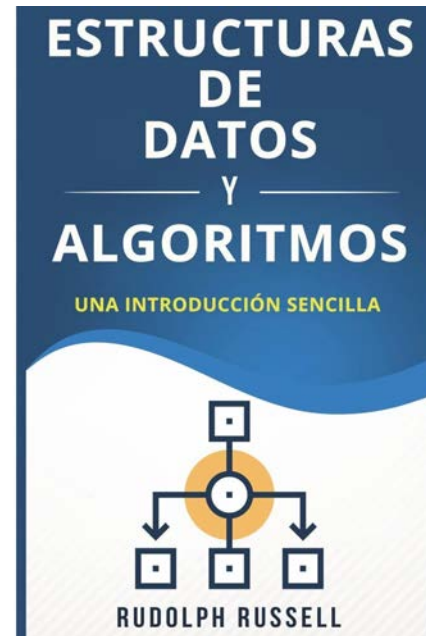
TIME TRAVEL



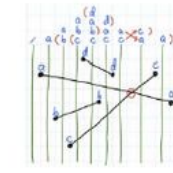
DYNAMIC GRAPHS



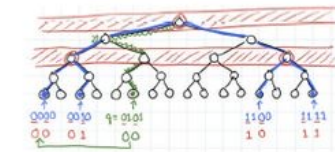
SUCCINCT



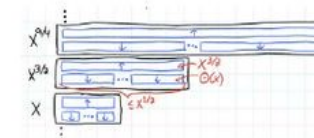
DYNAMIC OPTIMALITY



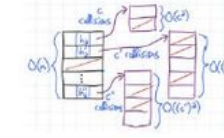
GEOMETRY



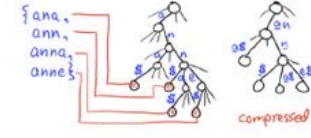
INTEGERS



MEMORY HIERARCHY



HASHING



STRINGS



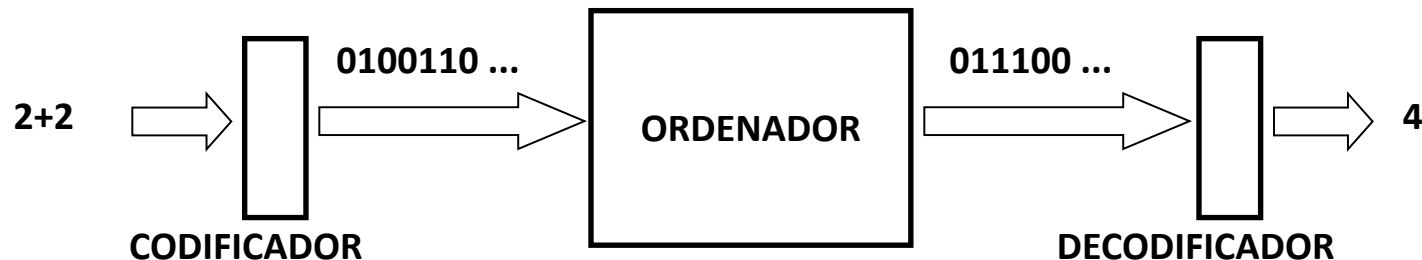


# Representación de la información



# Medida de la información

- En el interior del ordenador la información se almacena y procesa mediante un código que usa sólo dos valores, representados por el 0 y el 1, y que se denomina **código binario**.
- El ordenador codifica la información de entrada en código binario y decodifica la salida para presentar los resultados obtenidos:



- La unidad mínima de información es el **bit** (**B**inary **D**igi**T**). Cada bit puede tomar los valores 0 ó 1.
- La unidad con significado en informática es el **byte**, que son 8 bits.



# Códigos alfanuméricos

- Son códigos que permiten representar no sólo números sino también letras y otros símbolos.
- Información que necesitaría codificar:
  - las 26 letras del alfabeto
  - las 10 cifras decimales
  - los signos especiales (puntuación, punto y coma,...)
  - códigos de control,
  - el alfabeto griego, ...



# Código ASCII

Caracteres de control				Símbolos gráficos											
Nombre	Dec	Binario	Hex	Símbolo	Dec	Binario	Hex	Símbolo	Dec	Binario	Hex	Símbolo	Dec	Binario	Hex
NUL	0	0000000	00	space	32	0100000	20	@	64	1000000	40	`	96	1100000	60
SOH	1	0000001	01	!	33	0100001	21	A	65	1000001	41	a	97	1100001	61
STX	2	0000010	02	"	34	0100010	22	B	66	1000010	42	b	98	1100010	62
ETX	3	0000011	03	#	35	0100011	23	C	67	1000011	43	c	99	1100011	63
EOT	4	0000100	04	\$	36	0100100	24	D	68	1000100	44	d	100	1100100	64
ENQ	5	0000101	05	%	37	0100101	25	E	69	1000101	45	e	101	1100101	65
ACK	6	0000110	06	&	38	0100110	26	F	70	1000110	46	f	102	1100110	66
BEL	7	0000111	07	'	39	0100111	27	G	71	1000111	47	g	103	1100111	67
BS	8	0001000	08	(	40	0101000	28	H	72	1001000	48	h	104	1101000	68
HT	9	0001001	09	)	41	0101001	29	I	73	1001001	49	i	105	1101001	69
LF	10	0001010	0A	*	42	0101010	2A	J	74	1001010	4A	j	106	1101010	6A
VT	11	0001011	0B	+	43	0101011	2B	K	75	1001011	4B	k	107	1101011	6B
FF	12	0001100	0C	,	44	0101100	2C	L	76	1001100	4C	l	108	1101100	6C
CR	13	0001101	0D	-	45	0101101	2D	M	77	1001101	4D	m	109	1101101	6D
SO	14	0001110	0E	.	46	0101110	2E	N	78	1001110	4E	n	110	1101110	6E
SI	15	0001111	0F	/	47	0101111	2F	O	79	1001111	4F	o	111	1101111	6F
DLE	16	0010000	10	0	48	0110000	30	P	80	1010000	50	p	112	1110000	70
DC1	17	0010001	11	1	49	0110001	31	Q	81	1010001	51	q	113	1110001	71
DC2	18	0010010	12	2	50	0110010	32	R	82	1010010	52	r	114	1110010	72
DC3	19	0010011	13	3	51	0110011	33	S	83	1010011	53	s	115	1110011	73
DC4	20	0010100	14	4	52	0110100	34	T	84	1010100	54	t	116	1110100	74
NAK	21	0010101	15	5	53	0110101	35	U	85	1010101	55	u	117	1110101	75
SYN	22	0010110	16	6	54	0110110	36	V	86	1010110	56	v	118	1110110	76
ETB	23	0010111	17	7	55	0110111	37	W	87	1010111	57	w	119	1110111	77
CAN	24	0011000	18	8	56	0111000	38	X	88	1011000	58	x	120	1111000	78
EM	25	0011001	19	9	57	0111001	39	Y	89	1011001	59	y	121	1111001	79
SUB	26	0011010	1A	:	58	0111010	3A	Z	90	1011010	5A	z	122	1111010	7A
ESC	27	0011011	1B	;	59	0111011	3B	[	91	1011011	5B	{	123	1111011	7B
FS	28	0011100	1C	<	60	0111100	3C	\	92	1011100	5C		124	1111100	7C
GS	29	0011101	1D	=	61	0111101	3D	]	93	1011101	5D	}	125	1111101	7D
RS	30	0011110	1E	>	62	0111110	3E	^	94	1011110	5E	~	126	1111110	7E
US	31	0011111	1F	?	63	0111111	3F	_	95	1011111	5F	Del	127	1111111	7F



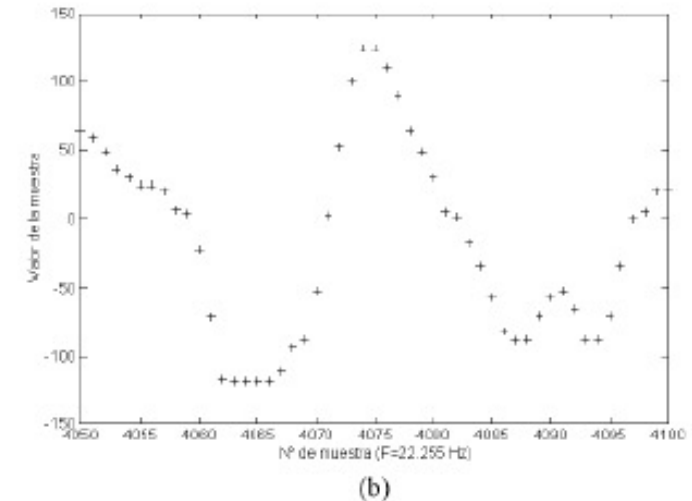
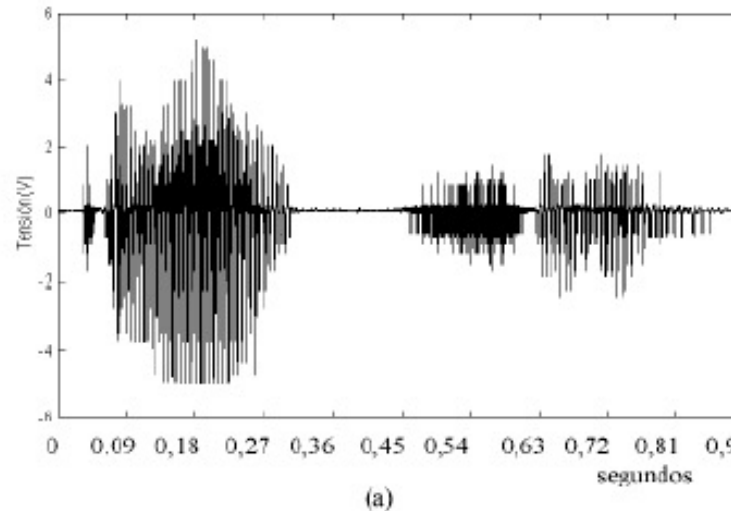
# Representación de información analógica

- En muchas aplicaciones un ordenador tiene entradas y/o salidas en forma analógica
- Una señal analógica es aquella que puede tomar cualquier valor dentro de un determinado **intervalo continuo**
- Paso de datos analógicos a digitales:
  - Conversores analógicos-digitales (ADC)
  - Toman muestras periódicas a intervalos fijos y convierten los valores a su equivalente digital
  - Este proceso recibe el nombre de *digitalización*



# Representación de sonido

- Se capta por medio de un micrófono que produce una señal analógica
- Simultáneamente al muestreo las muestras se digitalizan con un *conversor A/D*



Posición	contenido
1	2
2	2
3	2
4	2
.	.
4050	63
4051	58
4052	48
4053	35
4054	29
4055	24
4056	24

posición	contenido
4057	20
4058	6
4059	3
4060	-23
4061	-71
4062	-118
.	.
19996	1
19997	1
19998	1
19999	1
20000	1

(c)



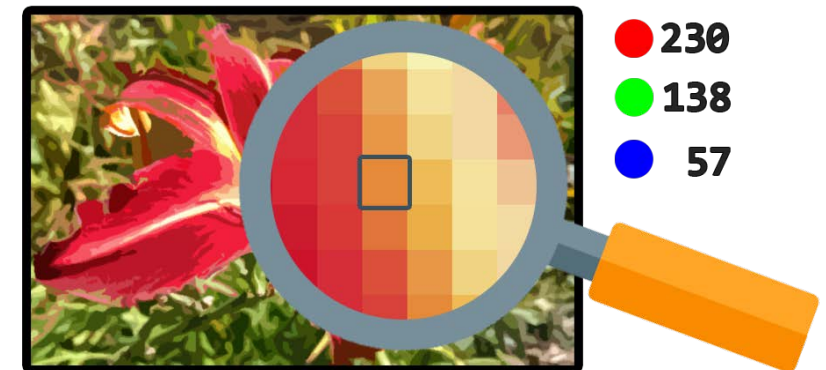
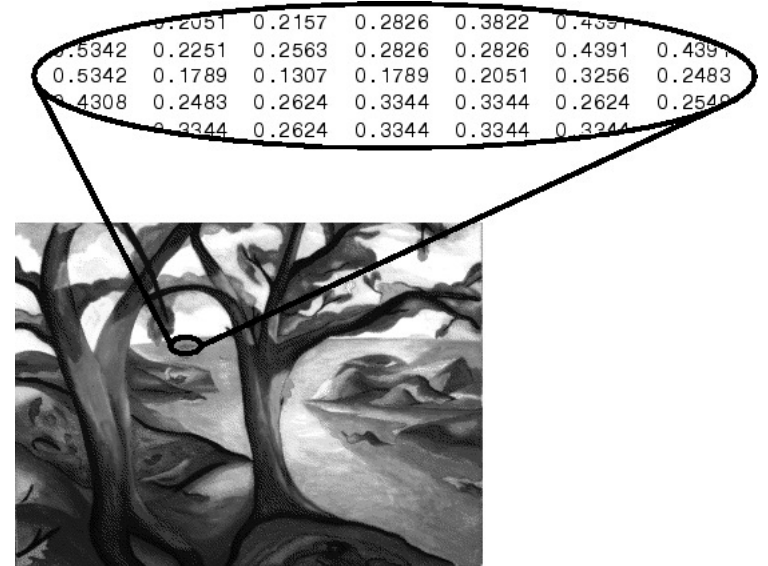


# Representación de imágenes

- Las imágenes se obtienen por periféricos como por ejemplo escáneres, cámaras digitales y cámaras de video
- Existen sistemas de codificación de imágenes muy diversos: BMP, TIFF, JPG, GIF, PNG, etc.
- Formas básicas de representar imágenes:
  - Mapas de bits
  - Mapas de vectores

# Representación de imágenes

- Mapas de bits:
  - Imagen compuesta por puntos
  - A cada punto se le asocia un atributo:
    - Nivel de gris (imagen en blanco y negro)
    - Nivel de color (imagen en color)
  - Para almacenar una imagen influyen 2 factores:
    - Número de puntos
    - Código de atributo (color) asociado a cada punto
  - Como no es posible almacenar infinitos puntos:
    - Se divide la imagen en una fina retícula (*elementos de imagen o pixels*)
    - A cada punto se le asigna como atributo el nivel de gris o color medio





# Algoritmos

procedimiento paso a paso para solucionar un problema

## PASOS:

- Análisis del problema
- Diseño del algoritmo
- Verificación manual
- Implementación en el computador mediante un PROGRAMA en algún lenguaje.



- Las características fundamentales que debe cumplir todo algoritmo son:
  - Debe ser **preciso** e indicar el orden de realización de cada paso
  - Debe estar **definido** (determinista): si se siguen los mismos pasos de un algoritmo n veces se debe obtener siempre el mismo resultado
  - Debe ser **finito**: si se siguen los pasos de un algoritmo se debe terminar en algún momento, es decir, tiene un número finito de pasos
- Para la descripción de un algoritmo se usan distintas representaciones como el lenguaje natural, el pseudocódigo, los árboles de decisión o los organigramas



- Exact instructions challenge:
  - [https://www.youtube.com/watch?v=cDA3\\_5982h8](https://www.youtube.com/watch?v=cDA3_5982h8)
- Ejemplo:
  - Descripción de un algoritmo, mediante lenguaje natural, para ir al cine a ver “BATMAN”

**Inicio**  
**Ver la cartelera de cines en el periódico**  
**Si proyectan BATMAN entonces**  
    **Ir al cine**  
    **Si hay cola entonces**  
        **Ponerse en ella**  
    **Mientras haya personas delante hacer**  
        **Avanzar en la cola**  
    **Si existen asientos disponibles entonces**  
        **Comprar una entrada**  
        **Encontrar el asiento correspondiente**  
        **Mientras proyectan la película**  
            **Ver la película**  
    **Sino (existen asientos disponibles)**  
        **Abandonar el cine**  
        **Volver a casa**  
**Sino (proyectan BATMAN)**  
    **Decidir otra actividad**  
**Fin**



# Programas

- Conjunto de instrucciones que se ejecutan en un ordenador:
  - Secuencialmente
  - En paralelo
- El ordenador maneja las instrucciones y datos representados como 0s o 1s (lenguaje máquina).
- Nosotros programamos un ordenador en lenguajes de más alto nivel que el máquina por comodidad, pero se requiere uno o más pasos para que el ordenador pueda utilizar nuestro programa.





# Niveles de los lenguajes



## Lenguaje alto nivel (Python)

suma = suma + numero

## Ensamblado

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

## Código máquina

A19401  
03063204  
A39401

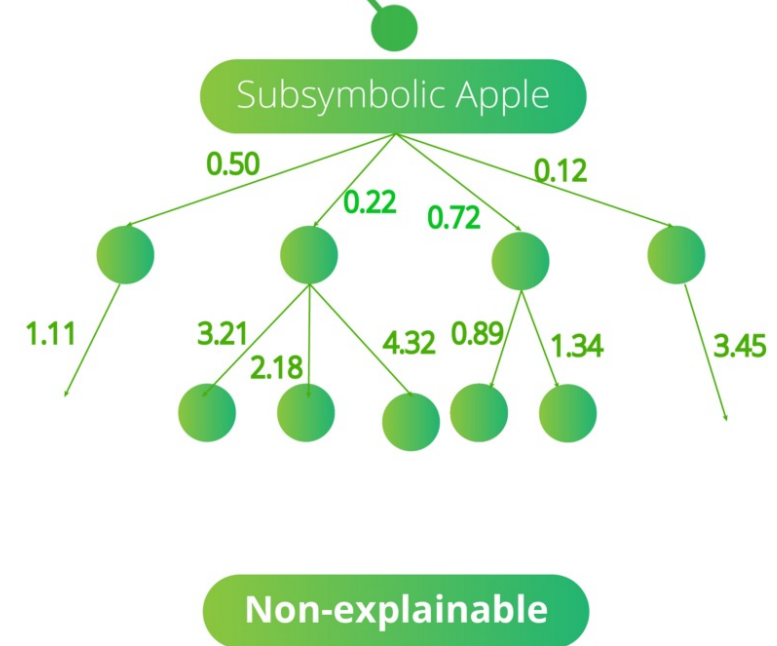
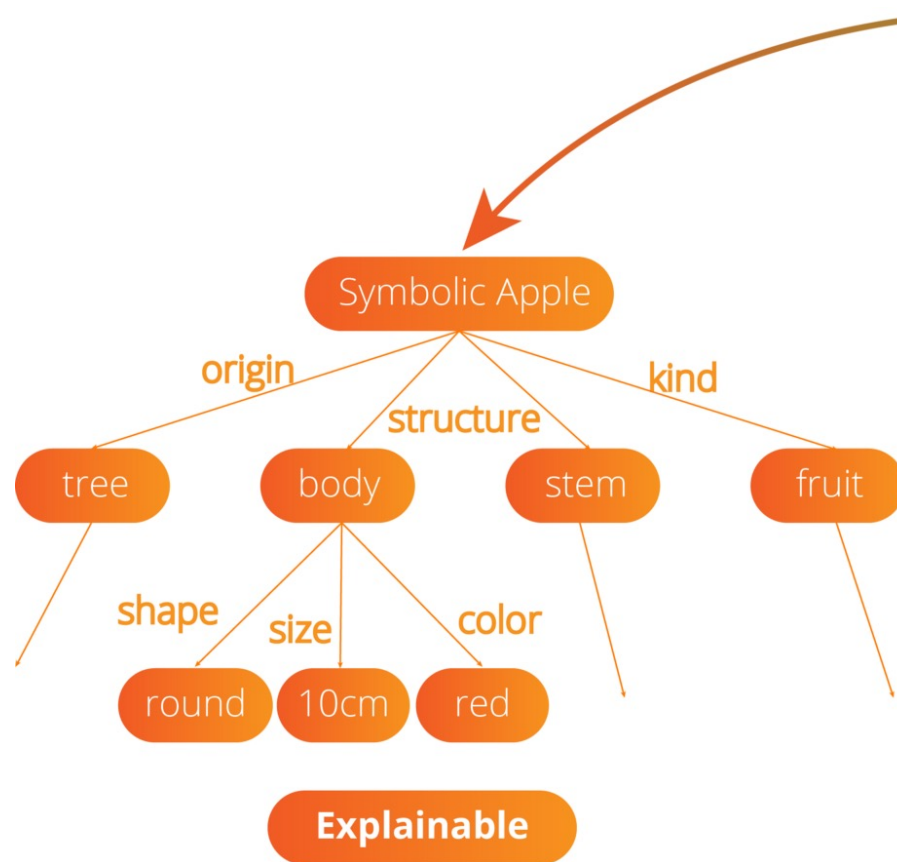


# Algoritmos en IA

- Un **programa sencillo** opera sobre una **representación simple**:
  - $\text{suma} = a + b$
- 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 en lenguaje



# Representación y razonamiento en IA



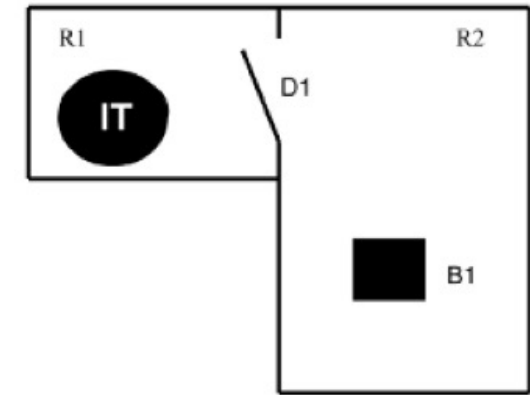


# Razonamiento en IA

- En informática, un sistema de razonamiento es un sistema software que **genera conclusiones a partir de los conocimientos disponibles**
- **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



# Razonamiento lógico



- Problema:
  - Un robot IT debe llegar de la habitación R1 a la R2
- Modelo de mundo:
  - Hay 3 tipos de elementos en el mundo: objetos\_movibles (robot, cajas), habitaciones y puertas

<code>INROOM(x, r)</code>	where <code>x</code> is an object of type <code>movable_object</code> , <code>r</code> is type <code>room</code>
<code>NEXTTO(x, t)</code>	where <code>x</code> is a <code>movable_object</code> , <code>t</code> is type <code>door</code> or <code>movable_object</code>
<code>STATUS(d, s)</code>	where <code>d</code> is type <code>door</code> , <code>s</code> is an enumerated type: <code>OPEN</code> or <code>CLOSED</code>
<code>CONNECTS(d, rx, ry)</code>	where <code>d</code> is type <code>door</code> , <code>rx</code> , <code>ry</code> are the room





# Razonamiento lógico

- Modelo de mundo para el estado inicial:

```
initial state:  
INROOM(IT, R1)  
INROOM(B1, R2)  
CONNECTS(D1, R1, R2)  
CONNECTS(D1, R2, R1)  
STATUS(D1, OPEN)
```

- Modelo de mundo para el estado objetivo:

```
goal state:  
INROOM(IT, R2)  
INROOM(B1, R2)  
CONNECTS(D1, R1, R2)  
CONNECTS(D1, R2, R1)  
STATUS(D1, OPEN)
```



- Tabla de diferencias:

operator	preconditions	add-list	delete-list
OP1: GOTODOOR (IT, dx)	INROOM (IT, rk) CONNECT (dx, rk, rm)	NEXTTO (IT, dx)	
OP2: GOTHRUDOOR (IT, dx)	CONNECT (dx, rk, rm) NEXTTO (IT, dx) STATUS (dx, OPEN) INROOM (IT, rk)	INROOM (IT, rm)	INROOM (IT, rk)

- La tarea más compleja es definir una función de evaluación para las **diferencias** entre los estados posibles
- La diferencia lógica entre el estado inicial y el objetivo sería:

$$\neg \text{INROOM}(\text{IT}, \text{R2}) \text{ or } \text{INROOM}(\text{IT}, \text{R2}) = \text{FALSE}$$



- Estado final resultante:

```
state after OP2:  
INROOM(IT, R2)  
INROOM(B1, R2)  
  
CONNECTS(D1, R1, R2)  
CONNECTS(D1, R2, R1)  
STATUS(D1, OPEN)  
NEXTTO(IT, D1)
```

- Se termina el proceso y se ejecuta el plan en orden inverso:  
***GOTODOOR(IT, D1), GOTHRUDDOOR(IT, D1)***



# Problemas del razonamiento lógico

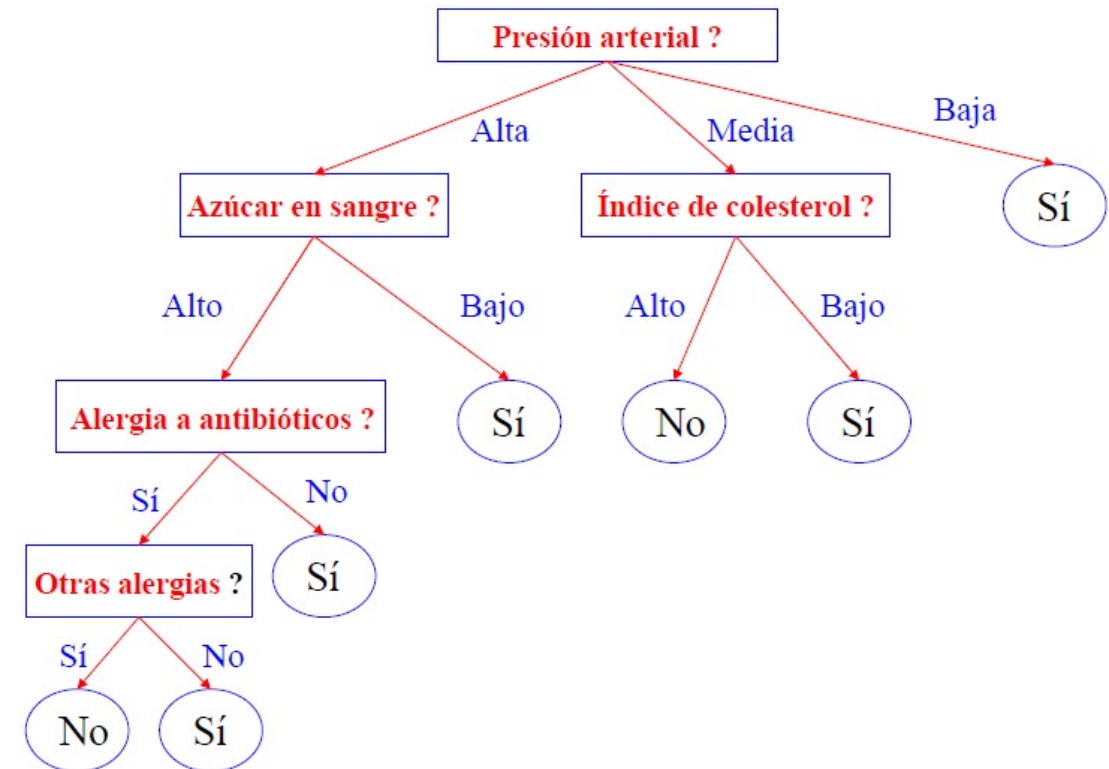
- Suposición de **mundo cerrado**:
  - El modelo de mundo contiene todo lo que el robot necesita saber
- El problema de representar un problema real de modo que sea tratable computacionalmente se denomina el **problema marco o problema de transducción**
- A mediados de los 80, los investigadores en robótica y los investigadores en IA tomaron caminos diferentes



# Árbol de decisión

- Un árbol de decisión es un mapa de los posibles resultados de una serie de decisiones relacionadas.

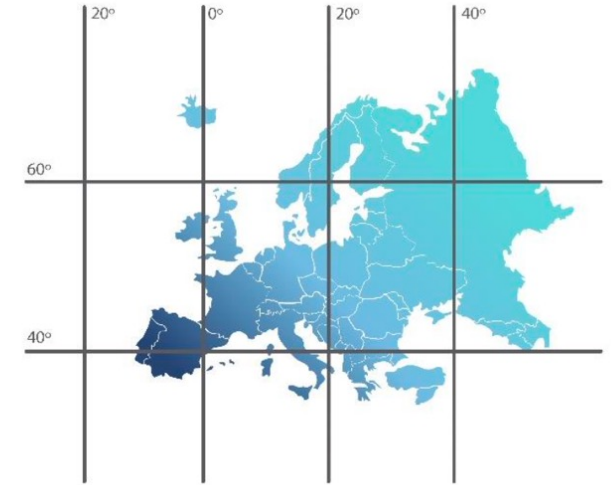
## Ejemplo: Árbol de decisión para “¿ Administrar fármaco F ?”





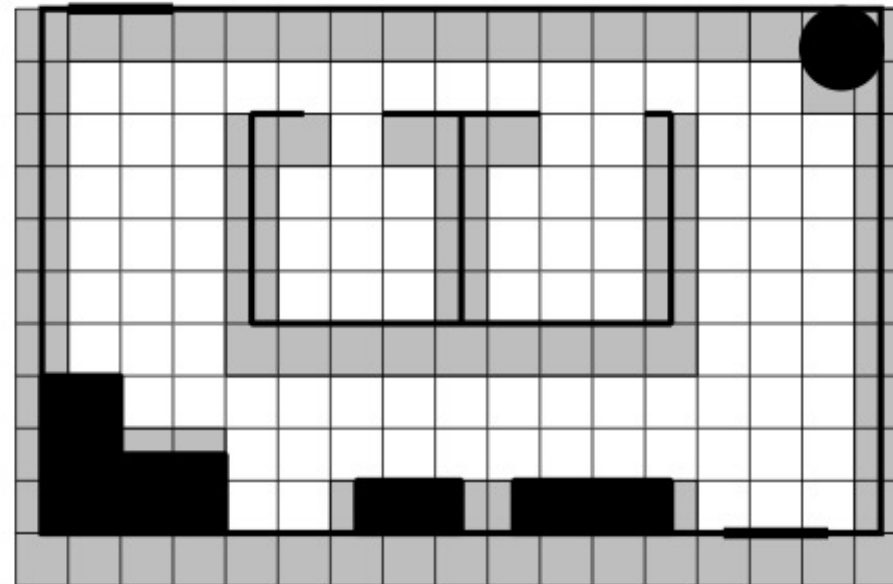
# Representación sub-simbólica

- Mapas métricos
  - Espacio bidimensional en el que los objetos se colocan con coordenadas precisas, por ejemplo, coordenadas cartesianas o geográficas
- Mapas topológicos
  - Solo consideran los lugares relevantes y las relaciones entre ellos. A menudo, también se almacenan las distancias entre los lugares. El mapa es entonces un grafo, en el que los nodos corresponden a los lugares y los arcos a los caminos

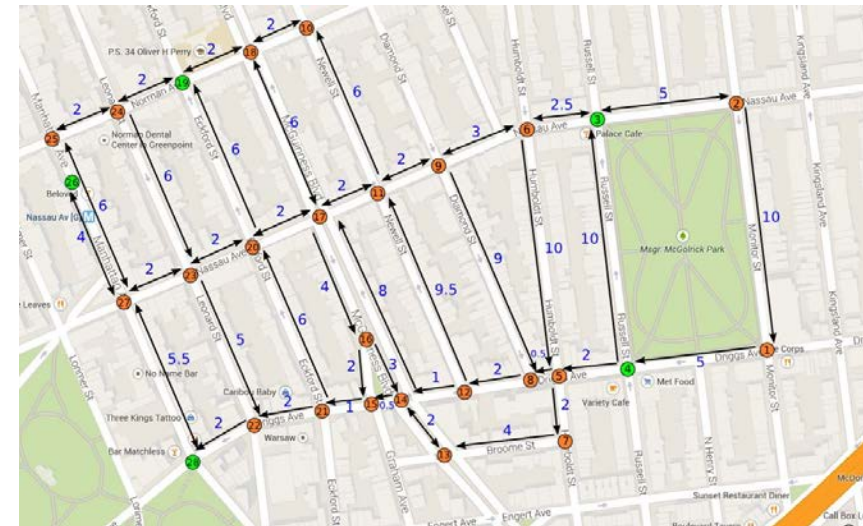
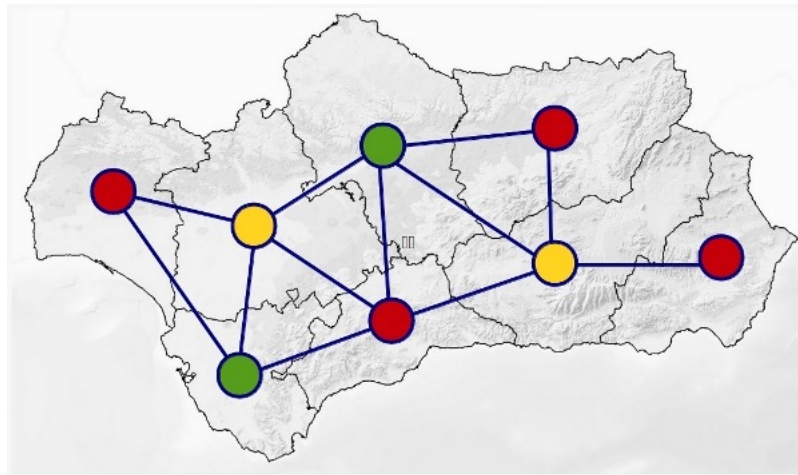




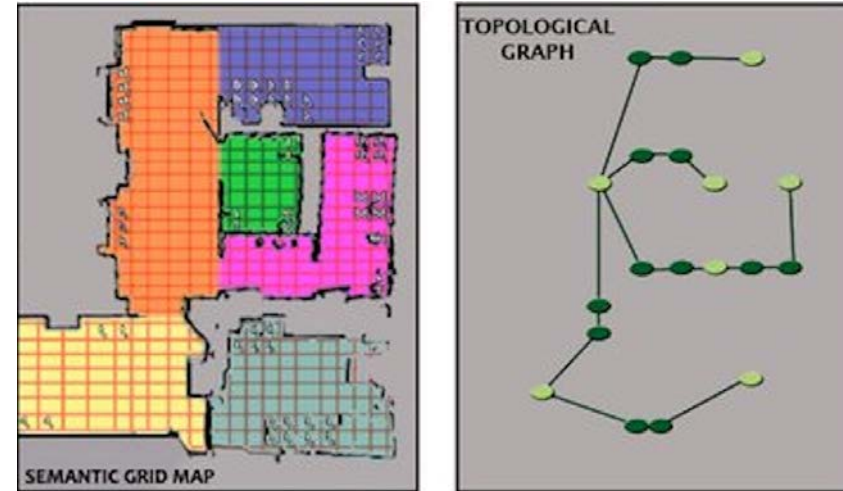
- Este método superpone un grid cartesiano 2D al entorno real
  - Si hay un objeto en el área contenida por un cuadrado del grid, el cuadrado se marca como ocupado
  - Se denominan **grids de ocupación**



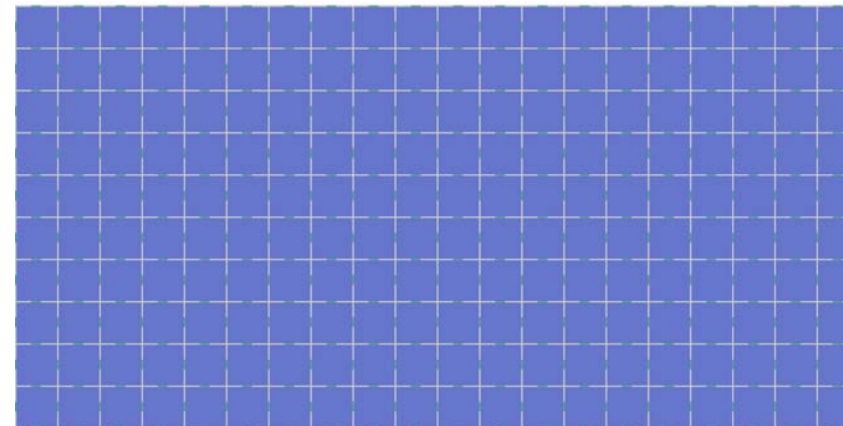
- Un grafo es un conjunto de objetos llamados vértices o nodos unidos por enlaces llamados aristas o arcos, que permiten representar relaciones binarias entre elementos de un conjunto.
- Los grafos permiten estudiar las interrelaciones entre unidades que interactúan unas con otras.



- Mapas topológicos
  - Búsqueda en grafo directamente



- Mapas métricos
  - Búsqueda en un grid
  - Búsqueda en un grafo



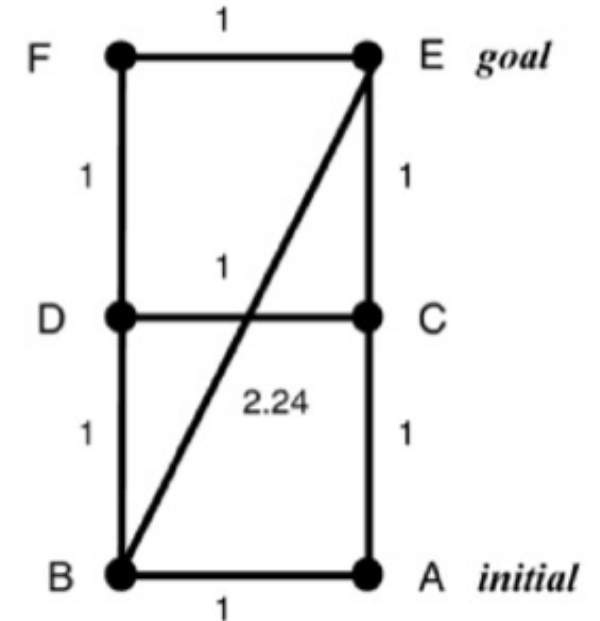


# Búsqueda en grafos

- Algoritmo A:  $f(n)=g(n)+h(n)$ 
  - $f(n)$  mide la bondad de moverse al nodo  $n$
  - $g(n)$  mide el coste de ir desde el nodo inicial hasta el nodo  $n$  (acumula)
  - $h(n)$  es el menor coste para ir desde  $n$  hasta el objetivo
  - Para conocer  $h(n)$  hay que visitar todos los nodos hasta el objetivo

$$f(B) = g(B) + h(B) = 1 + 2.24 = 3.24$$

$$f(C) = g(C) + h(C) = 1 + 1 = 2.0$$





# Algoritmos de búsqueda en grafos

- A**
  - A\* search algorithm
  - Alpha-beta pruning
  - Aperiodic graph
- B**
  - B\*
  - Barabási-Albert model
  - Belief propagation
  - Bellman-Ford algorithm
  - Bianconi-Barabási model
  - Bidirectional search
  - Borůvka's algorithm
  - Bottleneck traveling salesman problem
  - Breadth-first search
  - Bron-Kerbosch algorithm
  - Bully algorithm
- C**
  - Centrality
  - Chaitin's algorithm
  - Christofides algorithm
  - Clique percolation method
  - Closure problem
  - Color-coding
  - Contraction hierarchies
  - Courcelle's theorem
  - Cuthill-McKee algorithm
- D**
  - D\*
  - Degeneracy (graph theory)
  - Depth-first search
  - Dijkstra-Scholten algorithm
  - Dijkstra's algorithm
  - Dinic's algorithm
  - Disparity filter algorithm of weighted network
  - Distinctiveness centrality
  - Double pushout graph rewriting
  - DSatur
  - Dulmage-Mendelsohn decomposition
  - Dynamic connectivity
  - Dynamic link matching
- E**
  - Edmonds-Karp algorithm
  - Edmonds' algorithm
  - Blossom algorithm
  - Euler tour technique
  - External memory graph traversal
  - Extremal Ensemble Learning
- F**
  - FKT algorithm
  - Flooding algorithm
- G**
  - Floyd-Warshall algorithm
  - Force-directed graph drawing
  - Ford-Fulkerson algorithm
  - Fringe search
- H**
  - Hail-type theorems for hypergraphs
  - Havel-Hakimi algorithm
  - HCS clustering algorithm
  - Hierarchical closeness
  - Hierarchical clustering of networks
  - Hopcroft-Karp algorithm
- I**
  - Iterative deepening A\*
  - Initial attractiveness
  - Iterative compression
  - Iterative deepening depth-first search
- J**
  - Johnson's algorithm
  - Journal of Graph Algorithms and Applications
  - Jump point search
  - Junction tree algorithm
- K**
  - K shortest path routing
  - Karger's algorithm
  - KHOPCA clustering algorithm
  - Kleitman-Wang algorithms
  - Knight's tour
  - Knuth's Simpath algorithm
  - Kosaraju's algorithm
  - Kruskal's algorithm
- L**
  - LASCNN algorithm
  - Lexicographic breadth-first search
  - Link prediction
  - Longest path problem
- M**
  - MaxCliqueDyn maximum clique algorithm
- N**
  - METIS
  - Minimax
  - Minimum bottleneck spanning tree
  - Misra & Gries edge coloring algorithm
- P**
  - PageRank
  - Parallel all-pairs shortest path algorithm
  - Parallel breadth-first search
  - Path-based strong component algorithm
  - Pre-topological order
  - Prim's algorithm
  - Proof-number search
  - Push-relabel maximum flow algorithm
- R**
  - Reverse-delete algorithm
  - Rocha-Thatte cycle detection algorithm
- S**
  - Seidel's algorithm
  - Semantic Brand Score
  - Sethi-Ullman algorithm
  - Shortest Path Faster Algorithm
  - SMA\*
  - Spectral layout
  - Spreading activation
  - Stoer-Wagner algorithm
  - Subgraph isomorphism problem
  - Suurballe's algorithm
- T**
  - Tarjan's off-line lowest common ancestors algorithm
  - Tarjan's strongly connected components algorithm
  - Theta\*
  - Topological sorting
  - Transit node routing
  - Transitive closure
  - Transitive reduction
  - Travelling salesman problem
  - Tree traversal
- W**
  - Widest path problem
  - Wiener connector
- Y**
  - Yen's algorithm
- Z**
  - Zero-weight cycle problem



# Razonamiento probabilístico

- El razonamiento probabilístico se encarga de lidiar con la **incertidumbre** inherente de todo proceso de aprendizaje.
- Asigna **probabilidades a los sucesos** y la gestiona mediante **inferencia bayesiana**.
  - La inferencia bayesiana es un tipo de inferencia estadística en la que las evidencias u observaciones se emplean para actualizar o inferir la probabilidad de que una hipótesis pueda ser cierta.
  - El nombre «bayesiana» proviene del uso frecuente que se hace del teorema de Bayes durante el proceso de inferencia.





## ¿Cómo explicar representación a los estudiantes?



Un mapa es una representación de un lugar. Cualquier estudiante, independientemente de su edad, puede comprender que:

- Un mapa no es el territorio
- Los mapas abstraen muchos detalles
- Los mapas siguen ciertas convenciones de notación, como la forma en que se representan las carreteras o los edificios



## ¿Cómo explicar razonamiento a los estudiantes?



Siguiendo con el ejemplo anterior, una manera de entender el razonamiento es calcular el camino óptimo para llegar desde el punto A al B teniendo en cuenta las preferencias de los usuarios:

- Trayecto más rápido
- Trayecto más corto
- Trayecto que evite peajes
- ...





# Unidad AI+

41



Si no puede visionar el  
vídeo anterior, cliquea aquí:

<https://drive.google.com/file/d/1cKmi288hCdcMe5uZjPyXVXO4PMZK9Ugl/view?usp=sharing>

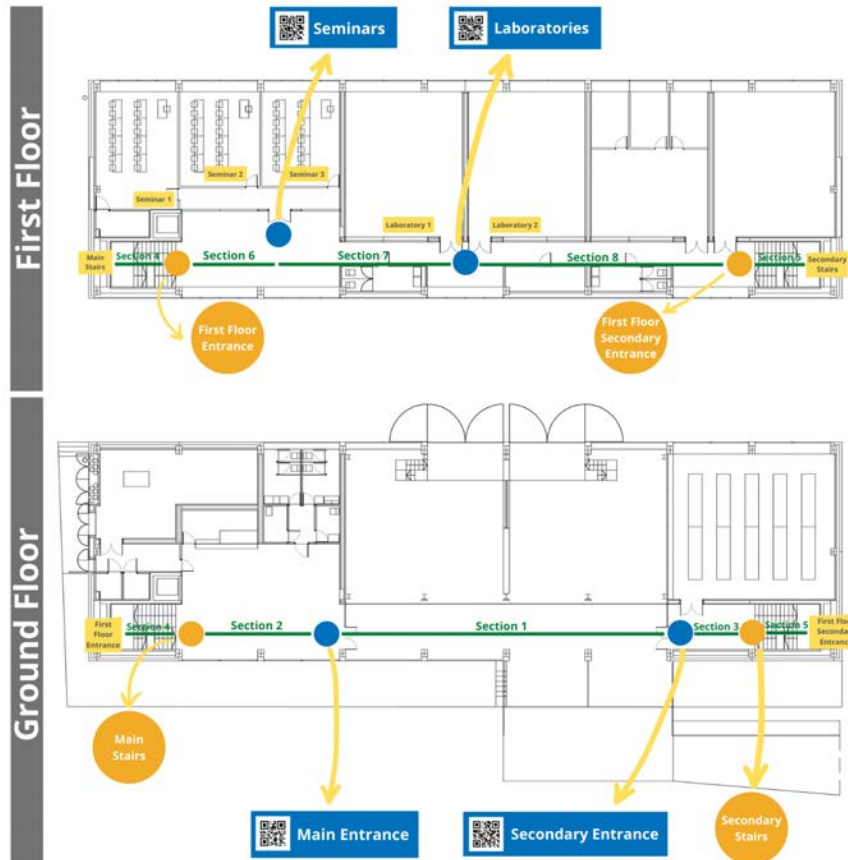


UNIVERSIDADE DA CORUÑA

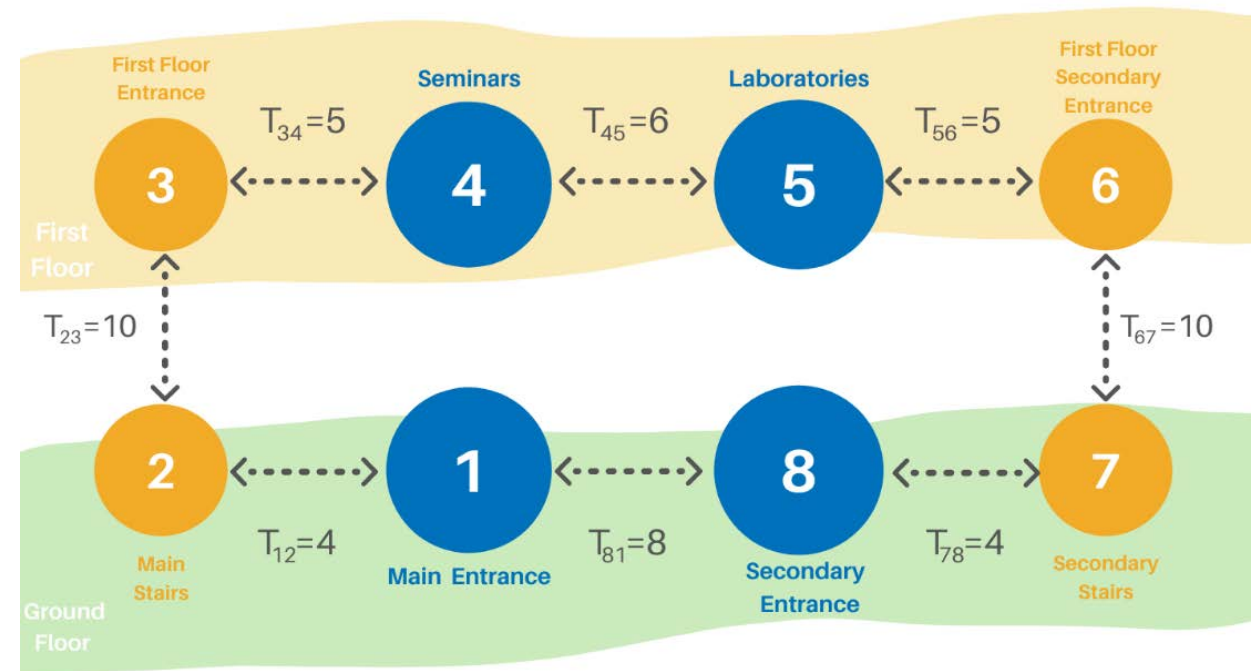


Co-funded by the  
Erasmus+ Programme  
of the European Union

## Paso 1: Definición de los puntos de interés



## Paso 2: Creación de un mapa topológico

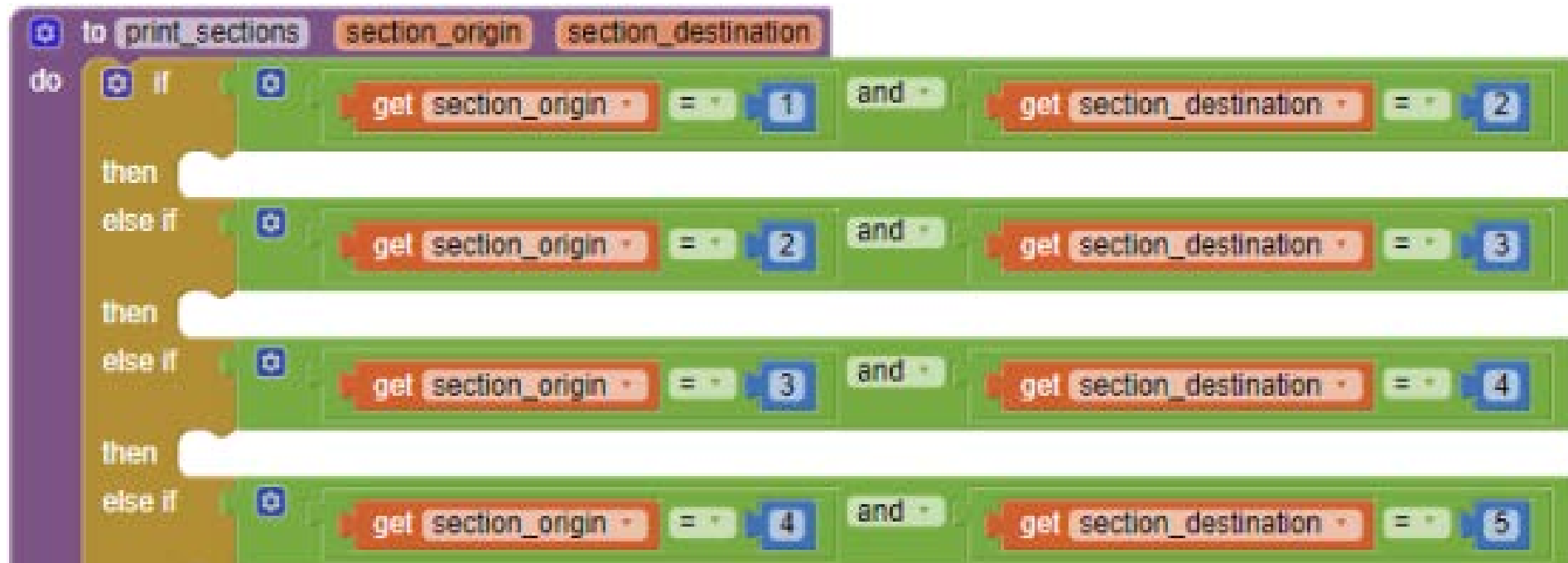




# Unidad AI+

43

Paso 3: Programación de la representación (en este caso, se representa mediante imágenes)





# Unidad AI+

44

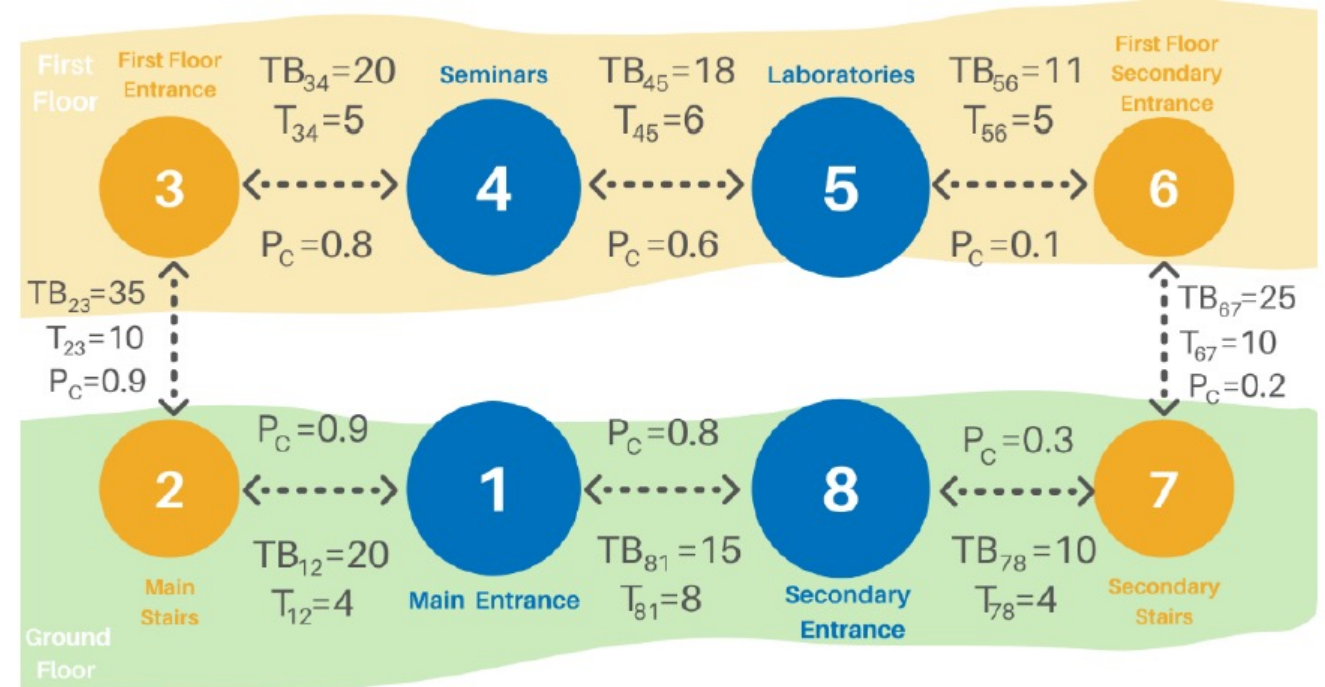
Paso 4: Cálculo de la ruta óptima (en este caso, el de menor distancia)

Main Entrance (1) → Secondary Entrance (8)		
	Path Sections	Distance
Route 1	Main Entrance (1) → Secondary Entrance (8)	8
	Total:	8
Route 2	Main Entrance (1) → Main Stairs (2)	4
	Main Stairs (2) → First Floor Entrance (3)	10
	First Floor Entrance (3) → Seminars (4)	5
	Seminars (4) → Laboratories (5)	6
	Laboratories (5) → First Floor Secondary Entrance (6)	5
	First Floor Secondary Entrance (6) → Secondary Stairs (7)	10
	Secondary Stairs (7) → Secondary Entrance (8)	4
	Total:	44



Paso 5: Cálculo de la ruta óptima pero ahora teniendo buscando el camino más rápido. Para ello, se considera que hay ciertos tramos que tienen mayor probabilidad de estar congestionados en hora punta (recreo)

LINK	1-2	2-3	3-4	4-5	5-6	6-7	7-8	8-1
$P_c$	0.9	0.9	0.8	0.6	0.1	0.2	0.3	0.5
T	4	10	5	6	5	10	4	8
TB	25	35	20	18	11	20	10	15





# Bibliografía

- La guía de recursos adaptados para el Bloque II – Parte III proporcionada como material adjunto a esta sesión contiene una relación exhaustiva de bibliografía adaptada
- Recursos prácticos recomendados:
  - **Elements of AI** modulo 2 y 3 <https://www.elementsofai.com/>
  - **Unidad 2 y 3 del proyecto AI+.** En la 2 se trabaja la parte de percepción y actuación y en la 3 representación y razonamiento pero todo sobre la misma app ([enlace](#))
  - **AI4k12**, iniciativa estadounidense <https://ai4k12.org/big-idea-2-overview/>



# Recursos prácticos

AI+ Unidad didáctica 3 (TU3)  
Árbol de decisión con App Inventor



## • ¿Cómo explicar representación a los estudiantes?



Un mapa es una representación de un lugar. Cualquier estudiante, independientemente de su edad, puede comprender que:

- Un mapa no es el territorio
- Los mapas abstraen muchos detalles
- Los mapas siguen ciertas convenciones de notación, como la forma en que se representan las carreteras o los edificios





# TU3: Razonamiento

## • ¿Cómo explicar razonamiento a los estudiantes?



Siguiendo con el ejemplo anterior, una manera de entender el razonamiento es calcular el camino óptimo para llegar desde el punto A al B teniendo en cuenta las preferencias de los usuarios:

- Trayecto más rápido
- Trayecto más corto
- Trayecto que evite peajes
- ...



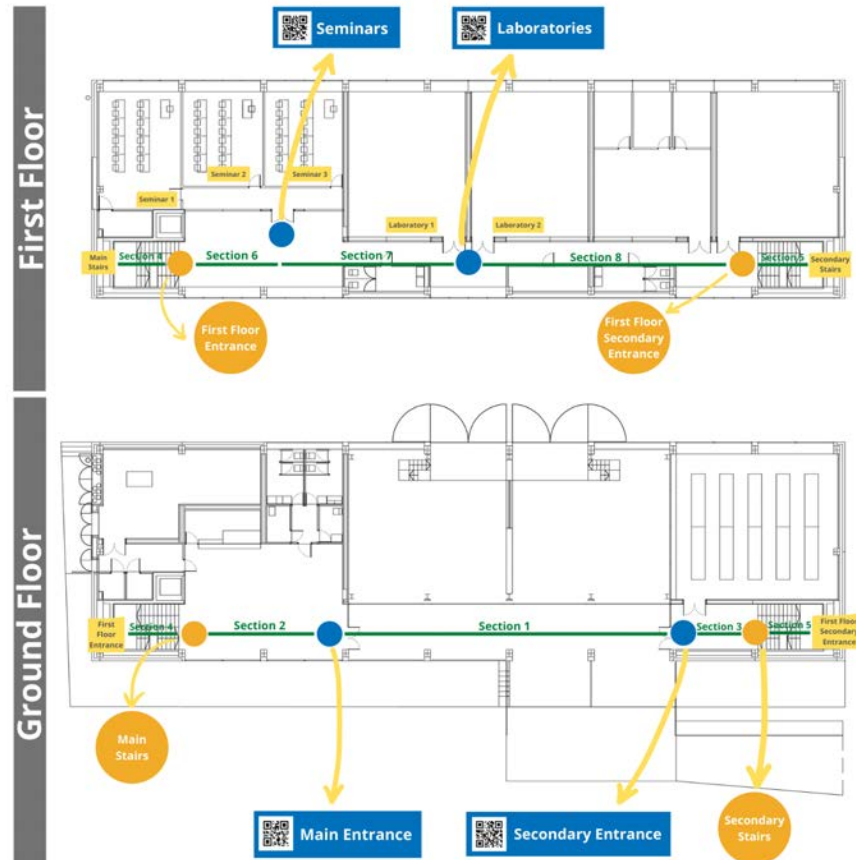
# TU3 AI+



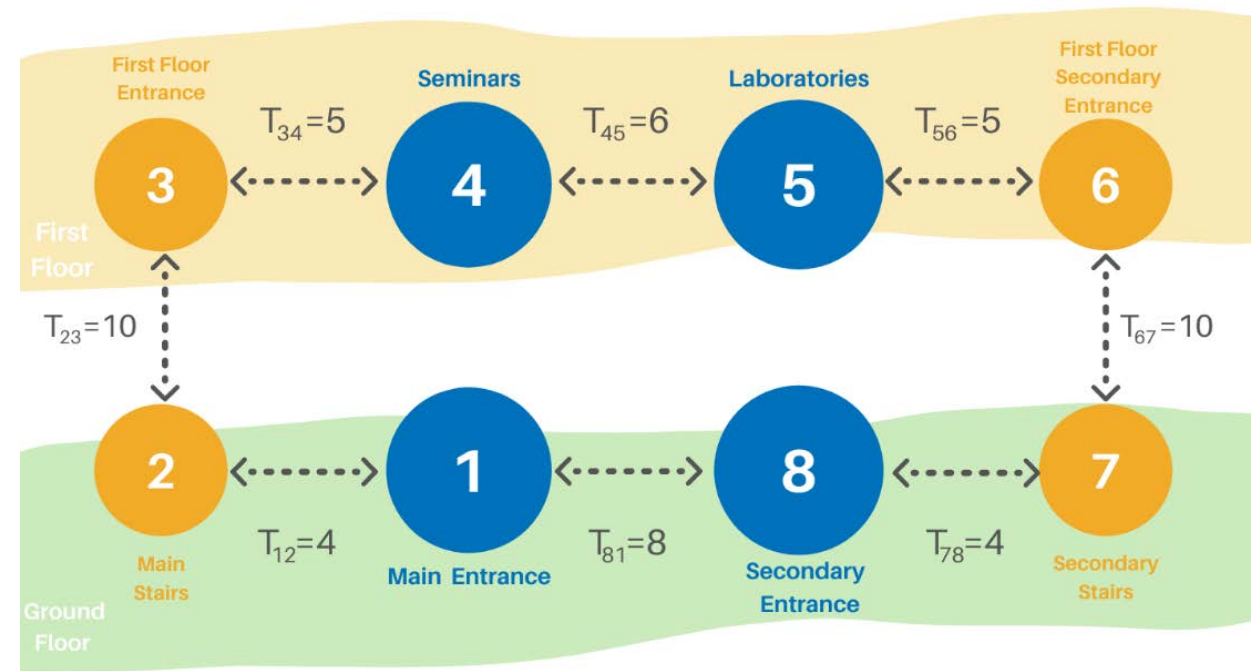
Si no puede visionar el vídeo anterior, cliquea aquí:

<https://drive.google.com/file/d/1cKmi288hCdcMe5uZjPyXVXO4PMZK9Ugl/view?usp=sharing>

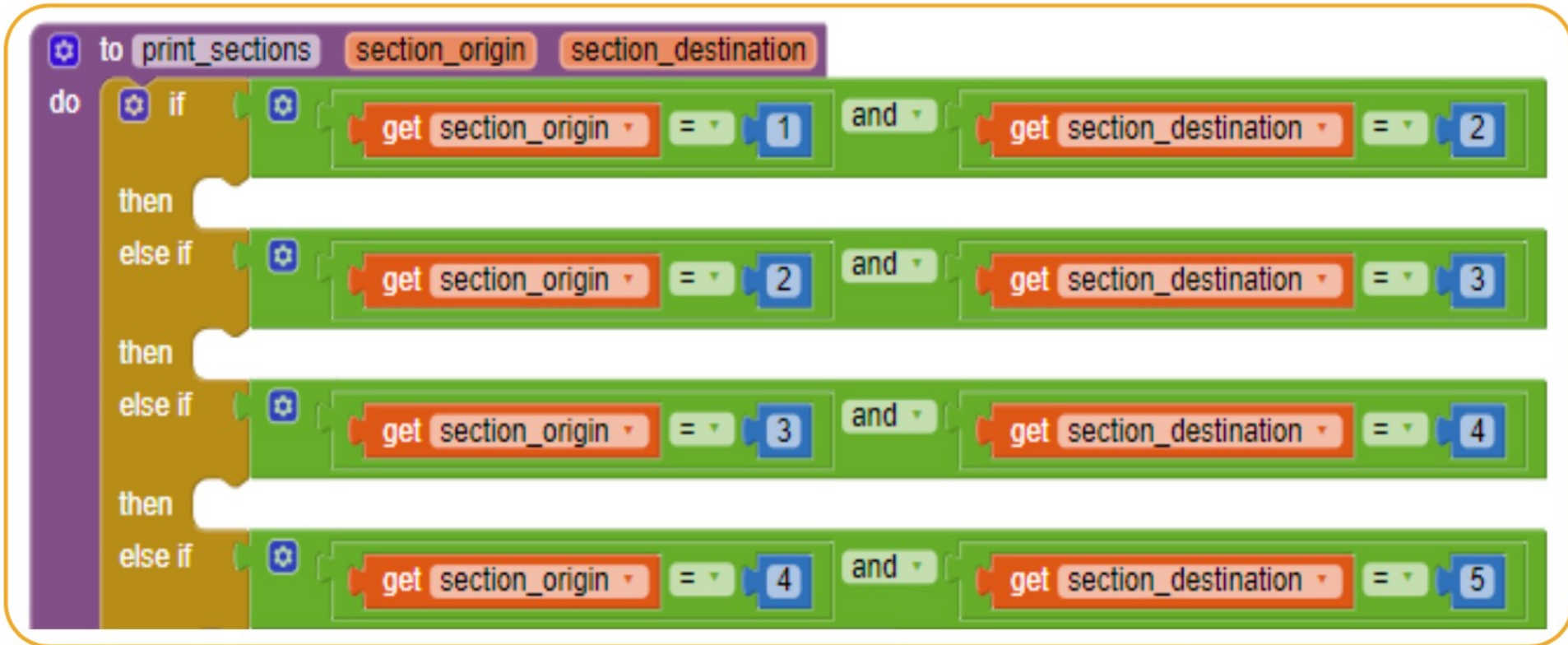
- Paso 1: Definición de los puntos de interés



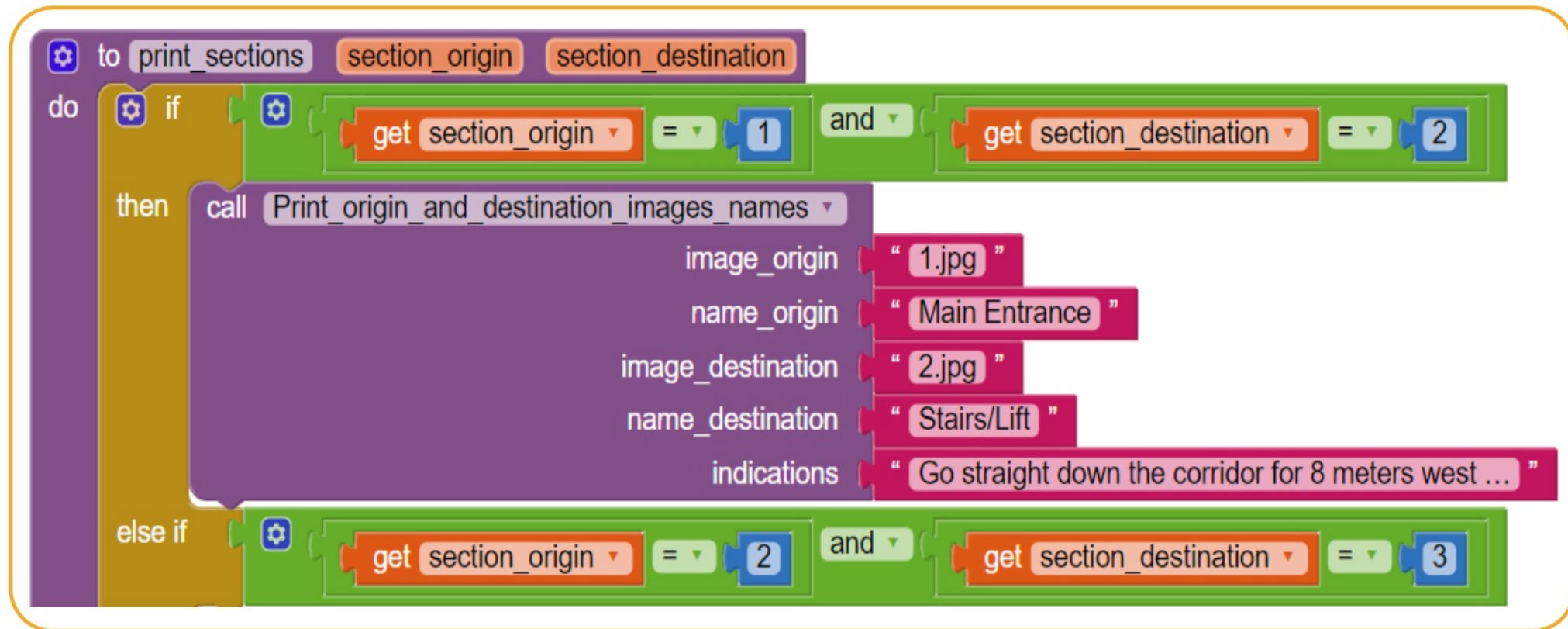
- Paso 2: Creación de un mapa topológico



- Paso 3: Programación de la representación (en este caso, se representa mediante imágenes)



- Paso 3: Programación de la representación (en este caso, se representa mediante imágenes)





- Paso 4: Cálculo de la ruta óptima (en este caso, el de menor distancia)

Main Entrance (1) → Secondary Entrance (8)		
	Path Sections	Distance
Route 1	Main Entrance (1) → Secondary Entrance (8)	8
	Total:	8
Route 2	Main Entrance (1) → Main Stairs (2)	4
	Main Stairs (2) → First Floor Entrance (3)	10
	First Floor Entrance (3) → Seminars (4)	5
	Seminars (4) → Laboratories (5)	6
	Laboratories (5) → First Floor Secondary Entrance (6)	5
	First Floor Secondary Entrance (6) → Secondary Stairs (7)	10
	Secondary Stairs (7) → Secondary Entrance (8)	4
	Total:	44



- Paso 4: Cálculo de la ruta óptima (en este caso, el de menor distancia)

Main Entrance (1) → Laboratories (5)		
	Path Sections	Distance
Route 1	Main Entrance (1) → Secondary Entrance (8)	8
	Secondary Entrance (8) → Secondary Stairs (7)	4
	Secondary Stairs (7) → First Floor Secondary Entrance (6)	10
	First Floor Secondary Entrance (6) → Laboratories (5)	5
	Total:	27
Route 2	Main Entrance (1) → Main Stairs (2)	4
	Main Stairs (2) → First Floor Entrance (3)	10
	First Floor Entrance (3) → Seminars (4)	5
	Seminars (4) → Laboratories (5)	6
	Total:	24



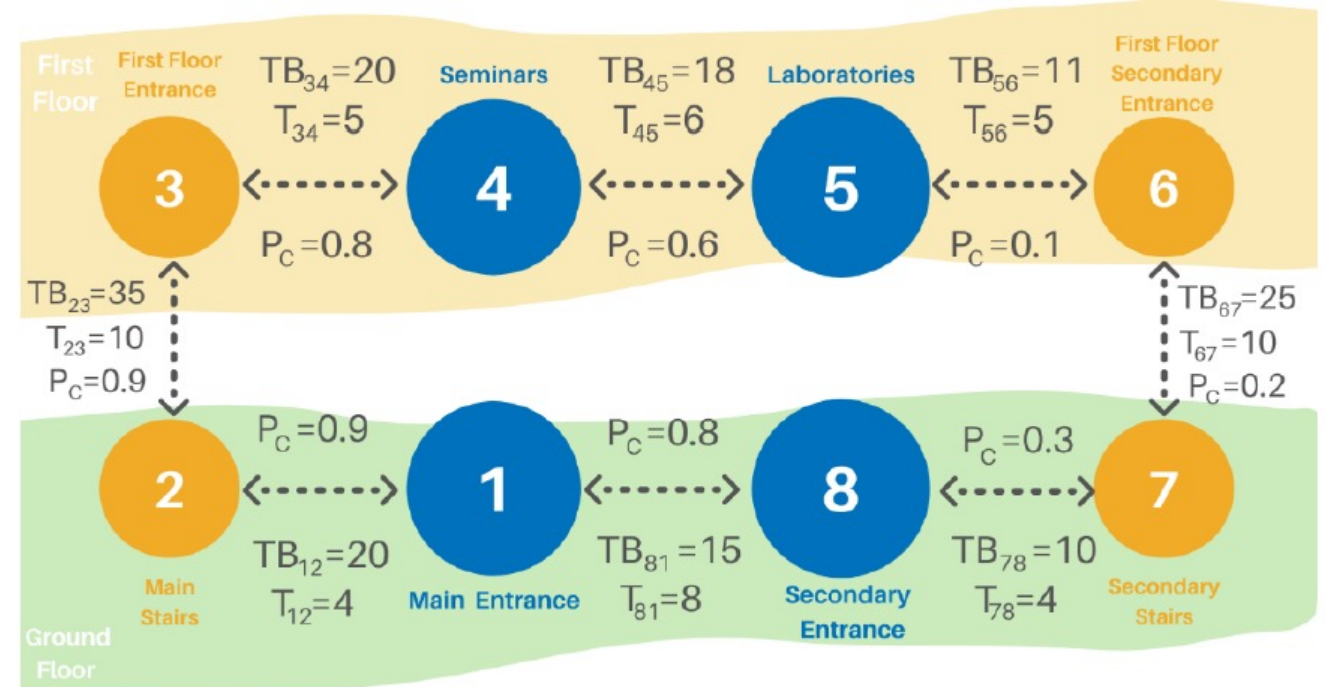
- Paso 4: Cálculo de la ruta óptima (en este caso, el de menor distancia)





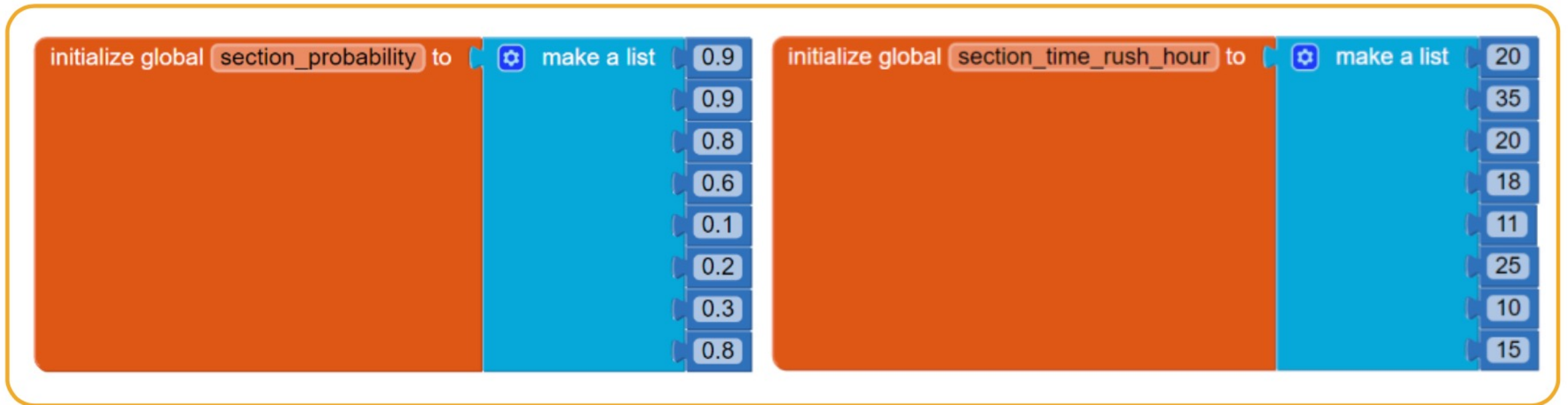
- Paso 5: Cálculo de la ruta óptima pero ahora teniendo buscando el camino más rápido. Para ello, se considera que hay ciertos tramos que tienen mayor probabilidad de estar congestionados en hora punta (recreo)

LINK	1-2	2-3	3-4	4-5	5-6	6-7	7-8	8-1
$P_c$	0.9	0.9	0.8	0.6	0.1	0.2	0.3	0.5
T	4	10	5	6	5	10	4	8
TB	25	35	20	18	11	20	10	15





- Paso 5: Cálculo de la ruta óptima pero ahora teniendo buscando el camino más rápido. Para ello, se considera que hay ciertos tramos que tienen mayor probabilidad de estar congestionados en hora punta (recreo)





# Adivina el animal

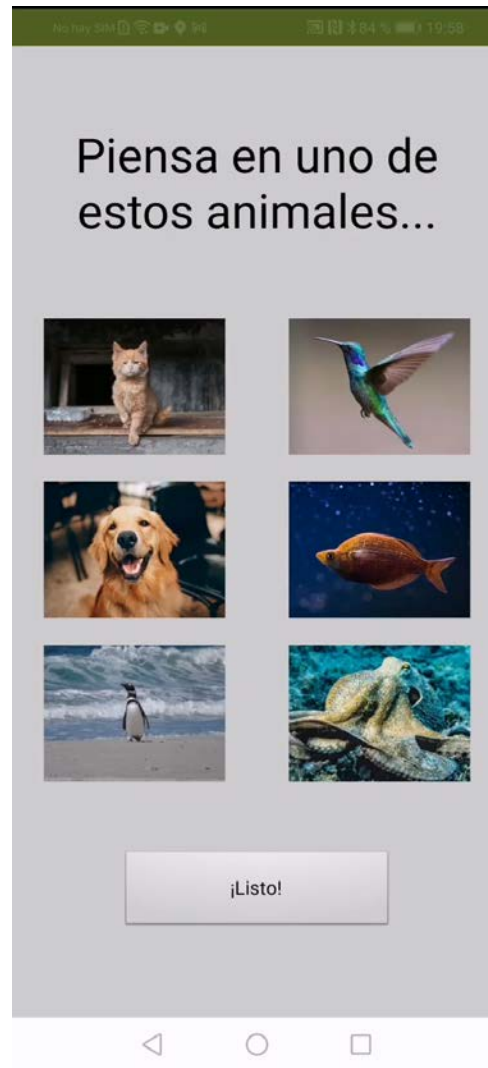
**Objetivo del juego:** adivinar el animal en el que está pensando una persona haciendo una serie de preguntas de sí o no.

**¿Cómo?:** 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



Si no puede visionar el vídeo anterior, cliquea aquí:

[https://drive.google.com/file/d/1sznV2XzYtkCTHrqB\\_8h6uBgK9G5kiwSF/view?usp=sharing](https://drive.google.com/file/d/1sznV2XzYtkCTHrqB_8h6uBgK9G5kiwSF/view?usp=sharing)





# Adivina el animal

Creamos la app en App Inventor.  
Para ello, proponemos la siguiente  
distribución en pantallas:

