Modelos de Inteligencia Artificial

Curso de Especialización de Inteligencia Artificial y Big Data
IES Gran Capitán 2023/24

UNIDAD 4.

Sistemas Expertos

Índice de contenidos

5. Representación y simulación de comportamientos básicos

Representación y simulación de comportamientos básicos

VENTAJAS

- Implementación rápida y sencilla.
- Maneja grandes cantidades de información simultáneamente.
- Toman decisiones de forma transparente.

DESVENTAJAS

- Muy difícil de diseñar un SE capaz de cubrir la totalidad de cierto campo o disciplina.
- No aportan soluciones imaginativas como un ser humano.
- No pueden tomar una decisión más allá de la información de la que disponen.

Representación y simulación de comportamientos básicos

A continuación, algunos de los frameworks y tecnologías más relevantes:

CLIPS Creado originalmente por la NASA, CLIPS está diseñado para facilitar la integración del conocimiento y el procesamiento de reglas. Soporta la programación en reglas, lógica orientada a objetivos y programación procedural.

Drools Es un motor de reglas de negocio (BRMS) y un sistema de gestión de procesos de negocio (BPMS) de código abierto para Java. Drools permite a los desarrolladores codificar reglas de negocio que son fácilmente actualizables y mantenibles.

Representación y simulación de comportamientos básicos

Jess Es un motor de reglas y scripting para Java inspirado en CLIPS. Jess soporta el desarrollo de sistemas expertos utilizando el paradigma de programación basada en reglas, permitiendo la creación de reglas complejas y su ejecución sobre datos estructurados.

Prolog Es un lenguaje de programación lógica asociado frecuentemente con la inteligencia artificial y la computación lingüística. Es utilizado para desarrollar sistemas expertos mediante la definición de hechos, reglas y metas..

Representación y simulación de comportamientos básicos

Expert System Shell Son marcos de trabajo o entornos que proporcionan una estructura básica para la construcción de sistemas expertos. Estas “cáscaras” vienen con herramientas para la definición de reglas, el motor de inferencia y a menudo interfaces para la adquisición del conocimiento y la explicación de las decisiones.

Pyke Es un motor de reglas de conocimiento para Python que combina la programación lógica y la programación orientada a objetos. Pyke permite a los desarrolladores crear sistemas expertos que pueden inferir conocimiento a partir de hechos definidos en bases de conocimiento.

Representación y simulación de comportamientos básicos

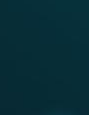
OpenCyc Es una base de conocimiento de código abierto y un motor de inferencia que implementa la ontología Cyc, proporcionando una rica base de conocimiento común para el desarrollo de aplicaciones de IA, incluyendo sistemas expertos.

PROLOG (Programation et logique)

Creado en los 70's por Alain Comerauer y Philippe Roussel.

Objetivo ⟶ crear una herramienta de programación cuya estructura fuera similar a la del lenguaje natural.

Programas en Prolog ⟶ base de datos ⟶ hechos / reglas ⟶ Se evalúan cuando se hace una consulta



Verdadero o Falso

PROLOG (Programation et logique)

Lenguaje de programación declarativo:

- basado en la lógica
- se describe el resultado final deseado en lugar de mostrar todos los pasos de trabajo

A un programa codificado en Prolog se le puede preguntar si un hecho es verdadero o si una declaración lógica es consistente con las reglas almacenadas en el mismo.

Manual de Prolog en SWISH

Ejercicio guiado en el aula

Realizar un programa pequeño en Prolog usando la herramienta online SWISH con datos:

```
% nombres de las personas
persona(alfonso).
persona(ana).
persona(carlos).
persona(carmen).
persona(fernando).
persona(isabel).
persona(luis).
persona(maria).
% deportes y su clasificación
tipo_deporte(marcha,atletismo).
tipo_deporte(vallas,atletismo).
tipo_deporte(cross,atletismo).
tipo_deporte(disco,atletismo).
tipo_deporte(martillo,atletismo).
tipo_deporte(salto,nieve).
tipo_deporte(esqui_alpino,nieve).
tipo_deporte(esqui_fondo,nieve).
tipo_deporte(snowboard,nieve).
tipo_deporte(trineo,nieve).
tipo_deporte(futbol,equipo).
tipo_deporte(baloncesto,equipo).
tipo_deporte(balonmano,equipo).
tipo_deporte(voleibol,equipo).
tipo_deporte(waterpolo,equipo).
```

```
% deporte que practica cada uno y nivel
practica_deporte(alfonso,categoria(baloncesto,nba)).
practica_deporte(ana,categoria(vallas,regional)).
practica_deporte(carlos,categoria(marcha,nacional)).
practica_deporte(carmen,categoria(futbol,tercera)).
practica_deporte(fernando,categoria(baloncesto,aficionado)).
practica_deporte(isabel,categoria(cross,regional)).
practica_deporte(luis,categoria(snowboard,aficionado)).
practica_deporte(maria,categoria(waterpolo,aficionado)).
```

Ejercicio guiado en el aula

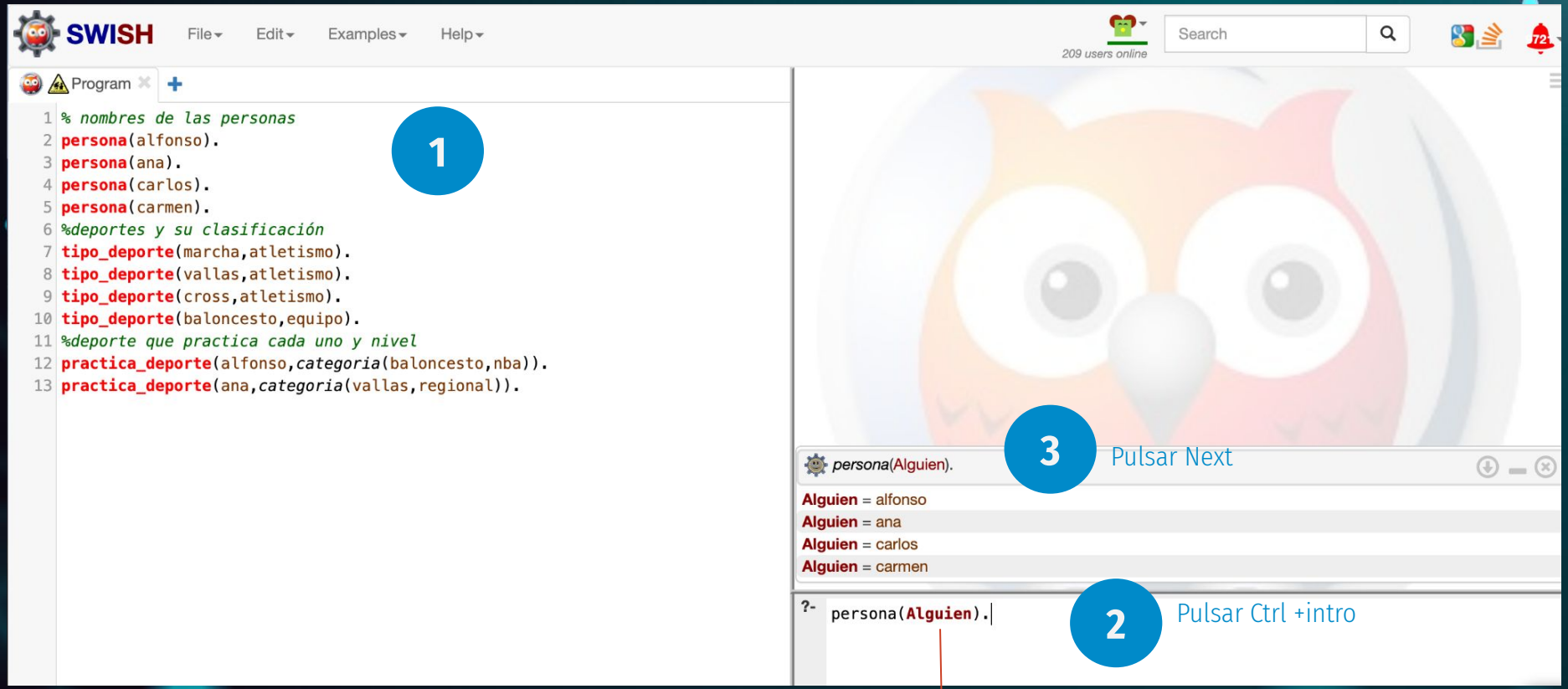
Tras introducir los datos anteriores, realizar las siguientes consultas:

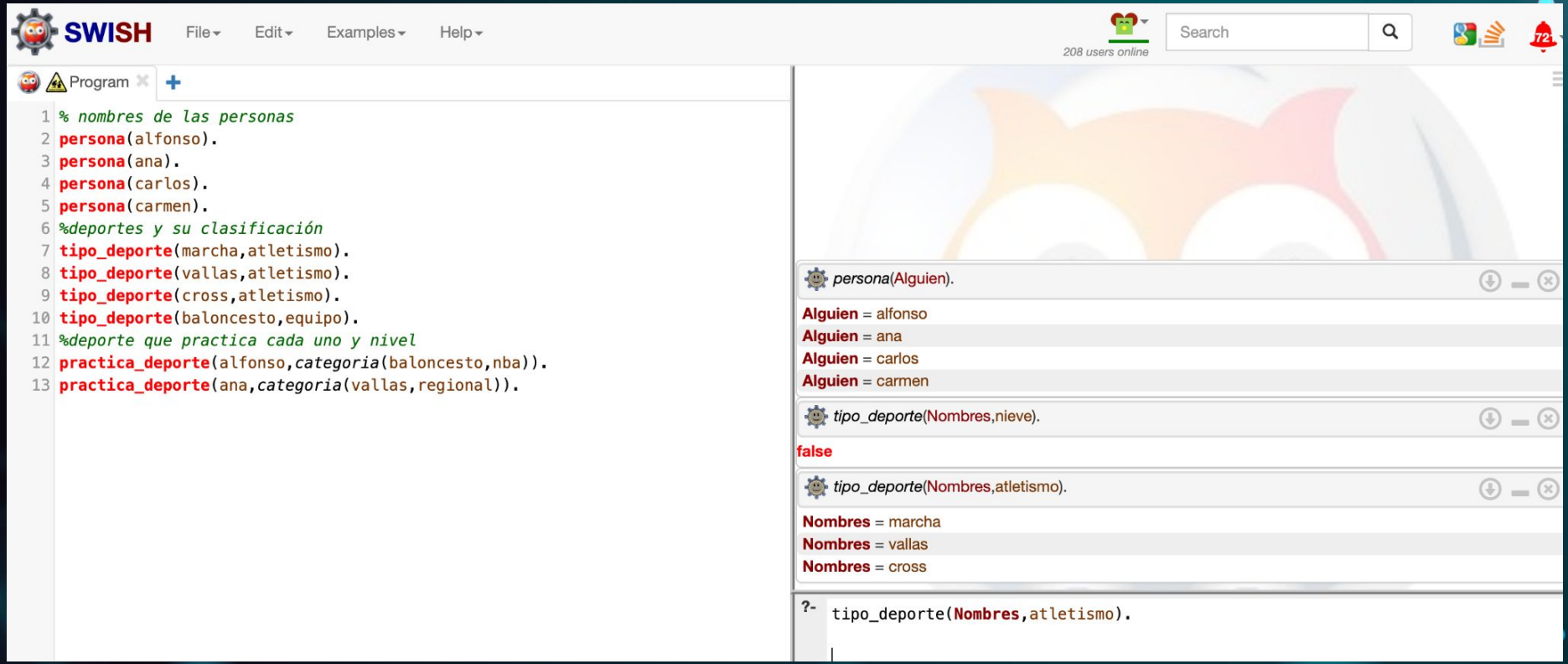
1. Preguntar por las personas → **persona(Alguien).**

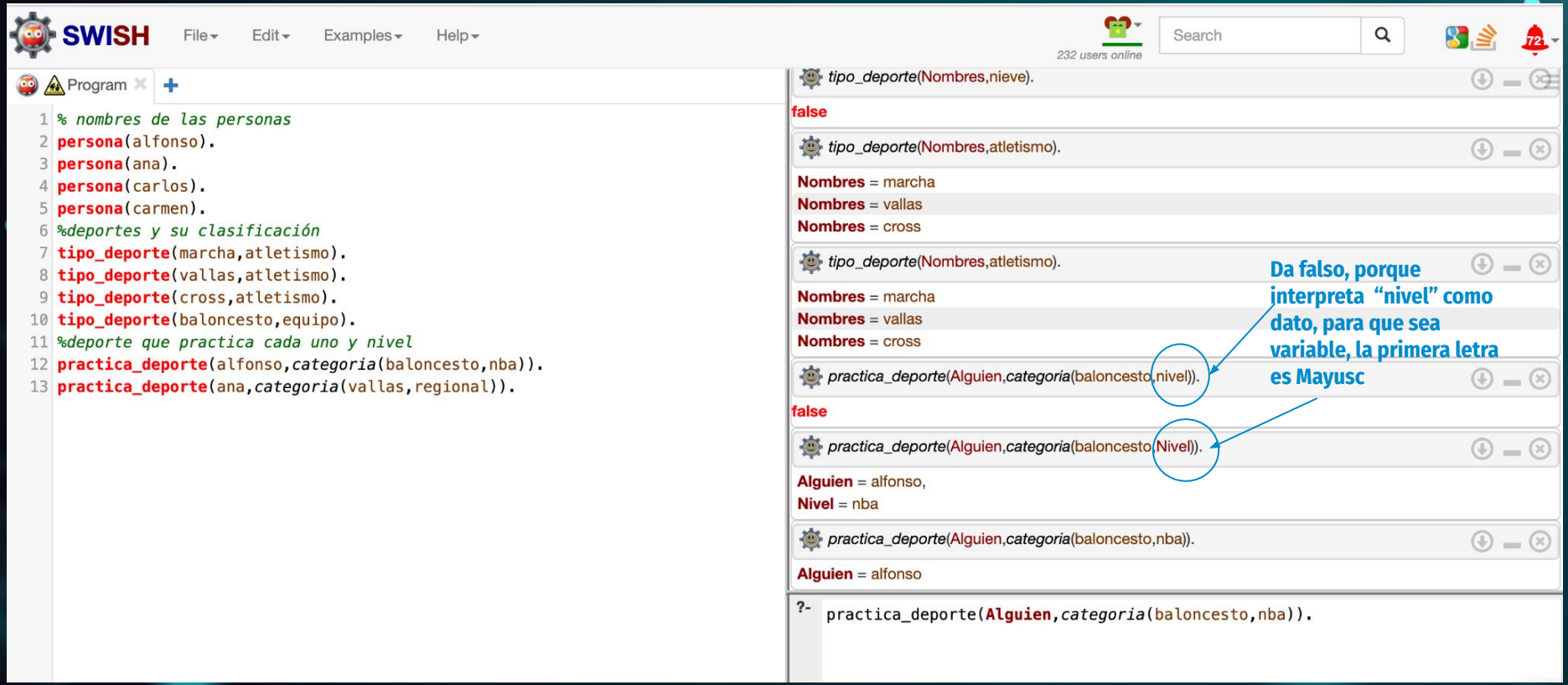
2. Consultar las personas que practican el deporte de nieve → **tipo_deporte(Nombres,nieve).**

3. Consultar las personas que practican baloncesto y muestra su nivel → **practica_deporte(Alguien,categoria(baloncesto,Nivel)).**

4. Consultar las personas que practican baloncesto y tienen el nivel de nba → **practica_deporte(Alguien,categoria(baloncesto,nba)).**


SWISH
File
Edit
Examples
Help
209 users online
Search
Program
1
% nombres de las personas
persona(alfonso).
persona(ana).
persona(carlos).
persona(carmen).
%deportes y su clasificación
tipo_deporte(marcha,atletismo).
tipo_deporte(vallas,atletismo).
tipo_deporte(cross,atletismo).
tipo_deporte(baloncesto,equipo).
%deporte que practica cada uno y nivel
practica_deporte(alfonso,categoria(baloncesto,nba)).
practica_deporte(ana,categoria(vallas,regional)).
3
Pulsar Next
persona(Alguien).
Alguien = alfonso
Alguien = ana
Alguien = carlos
Alguien = carmen
?-
persona(Alguien).
2
Pulsar Ctrl +intro
Variable


SWISH
File
Edit
Examples
Help
208 users online
Search
72
Program
% nombres de las personas
persona(alfonso).
persona(ana).
persona(carlos).
persona(carmen).
%deportes y su clasificación
tipo_deporte(marcha,atletismo).
tipo_deporte(vallas,atletismo).
tipo_deporte(cross,atletismo).
tipo_deporte(baloncesto,equipo).
%deporte que practica cada uno y nivel
practica_deporte(alfonso,categoria(baloncesto,nba)).
practica_deporte(ana,categoria(vallas,regional)).
persona(Alguien).
Alguien = alfonso
Alguien = ana
Alguien = carlos
Alguien = carmen
tipo_deporte(Nombres,nieve).
false
tipo_deporte(Nombres,atletismo).
Nombres = marcha
Nombres = vallas
Nombres = cross
?-
tipo_deporte(Nombres,atletismo).


SWISH
File
Edit
Examples
Help
232 users online
Search
Program
1 % nombres de las personas
2 persona(alfonso).
3 persona(ana).
4 persona(carlos).
5 persona(carmen).
6 %deportes y su clasificación
7 tipo_deporte(marcha,atletismo).
8 tipo_deporte(vallas,atletismo).
9 tipo_deporte(cross,atletismo).
10 tipo_deporte(baloncesto,equipo).
11 %deporte que practica cada uno y nivel
12 practica_deporte(alfonso,categoria(baloncesto,nba)).
13 practica_deporte(ana,categoria(vallas,regional)).
tipo_deporte(Nombres,nieve).
false
tipo_deporte(Nombres,atletismo).
Nombres = marcha
Nombres = vallas
Nombres = cross
tipo_deporte(Nombres,atletismo).
Nombres = marcha
Nombres = vallas
Nombres = cross
practica_deporte(Alguien,categoria(baloncesto,nivel)).
false
practica_deporte(Alguien,categoria(baloncesto,Nivel)).
Alguien = alfonso,
Nivel = nba
practica_deporte(Alguien,categoria(baloncesto,nba)).
Alguien = alfonso
?- practica_deporte(Alguien,categoria(baloncesto,nba)).
Da falso, porque interpreta "nivel" como dato, para que sea variable, la primera letra es Mayusc

¿Qué falta en este ejemplo para que cumpla con la principal característica de un Sistema Experto?

La estructura lógica **<<SI …. ENTONCES>>**

En Prolog esta estructura se representa de la siguiente forma:

Partiendo de un ejemplo en el que se pretende clasificar perros en función de sus características, para:

SI (tamaño es pequeño Y orejas son grandes) ENTONCES perro ES chihuahua

En Prolog sería:

perro(chihuahua) :- tamano(pequeno), orejas(grandes).

Dos puntos y un guión es el **SI**

La coma es el conector lógico **Y**, y el punto y coma es el conector **O**

Otro ejemplo:

```
se_llevan_bien(juan,pedro).

se_llevan_bien(luis,ana).

se_llevan_bien(maria,ana).

se_llevan_bien(Cualquiera,pedro).

se_llevan_bien(pedro,Cualquiera).
```

Consultas:

1. se_lleva_bien(pedro,maria) ⟶ true
2. se_lleva_bien(juan,maria) ⟶ false