

INSTITUTO POLITECNICO NACIONAL



ESCUELA SUPERIOR DE CÓMPUTO

FUNDAMENTOS DE INTELIGENCIA ARTIFICIAL

Prof. Hernández Cruz Macario

PRÁCTICA 06

4BM1

Corona Reyes Mauricio Dassel

Martinez Méndez Diego

Pacheco Sánchez Rodrigo

Ingeniería en Inteligencia Artificial

Resumen:

Nuestra práctica consiste en desarrollar un sistema experto con información de distintas especies de animales, con el fin de consultar e imprimir dicha información empleando reglas establecidas en un archivo prolog. Nuestro programa fue desarrollado empleando las herramientas de la biblioteca de Python, Tkinter para desarrollar la interfaz de usuario y que la experiencia con nuestro programa fuera lo más cómoda y sencilla posible.

Introducción:

En el presente trabajo desarrollamos un programa empleando el lenguaje de programación Python y Prolog para la elaboración de un sistema experto combinando ambas tecnologías.

Prolog es un lenguaje de programación basado en el paradigma de programación lógica, asociado en muchas ocasiones con la inteligencia artificial y el lenguaje computacional, el nombre de este lenguaje de programación proviene de combinar las palabras "PROgramación" y "LÓGica", prolog.

Se trata de un lenguaje de programación declarativo, funciona con base en hechos y reglas para dar sentido y "lógica" a su razonamiento en la ejecución de programas a partir de relaciones lógicas.

Los sistemas expertos según la definición del profesor Edward Feigenbaum, se tratan de programas inteligentes para ordenadores que emplean conocimiento humano y procedimientos inferenciales en la resolución de problemas.

Los sistemas expertos están compuestos por hechos y heurísticas, donde los hechos se tratan de cuerpos de información, mientras que los heurísticos son reglas de razonamiento plausible en las que se basa el sistema para obtener una solución al problema planteado.

Para nuestro programa, nosotros hemos ingresado tanto los hechos como las heurísticas dentro de nuestro programa prolog, para facilitar el razonamiento de la consulta creamos reglas de inferencia basadas en los hechos, finalmente nuestro código en Python obtendrá el tipo de especie que el usuario ingrese y obtendrá su información a partir del programa prolog.

Desarrollo:

Comenzando con la descripción de nuestro programa, elegimos el lenguaje de programación python para desarrollar nuestra práctica y utilizamos la biblioteca Tkinter para poderlo implementar, puesto que ofrece diversas herramientas que facilitaron la elaboración de nuestro programa y su interfaz gráfica.

A continuación, presentamos las reglas de inferencia propuestas en nuestro archivo prolog así como la información ingresada en el mismo archivo que servirá para que nuestro sistema experto busque a través del nombre científico del animal y pueda mostrar la información más adelante en nuestro programa Python.

```
• • •
frame(elephantidae, subclase_de(proboscidea), propiedades([son(elefantes)]).
frame(equidae, subclase_de(perissodactyla), propiedades([son(compatibles)]).
frame(rhinocerrotidae, subclase_de(perissodactyla), propiedades([tien(cuerno]]).
frame(noctilionoidea, subclase_de(chiroptera), propiedades([son(soss]])).
frame(noctilionoidea, subclase_de(chiroptera), propiedades([se(pescador)])).
frame(animal, subclase_de(objeto), propiedades([puede(sentir), puede(respirar)])).
frame(aminfero, subclase_de(animal), propiedades([puede(mamar), respira(aire]])).
frame(artiodactilo, subclase_de(animifero), propiedades([tiene(pesugnas_pares), comen(plantas)])).
frame(arnivora, subclase_de(mamifero), propiedades([tiene(cerebro_desarrollado)])).
frame(primates, subclase_de(mamifero), propiedades([tiene(cerebro_desarrollado)])).
frame(perissodactyla, subclase_de(mamifero), propiedades([tiene(pesugnas_impares)])).
frame(chiroptera, subclase_de(mamifero), propiedades([tien(alas), sefroedor)])).
frame(camelidos, subclase_de(artiodactilo), propiedades([fuel(alas), sefroedor)])).
frame(suidae, subclase_de(artiodactilo), propiedades([son(grande]s)])).
frame(suidae, subclase_de(artiodactilo), propiedades([son(grandes_s_simlos)])).
frame(felidae, subclase_de(carnivora), propiedades([son(grandes_s_simlos)])).
frame(felidae, subclase_de(carnivora), propiedades([son(filenos)])).
 imagen('elefante.png')])).
frame(caballo, subclase_de(equidae), propiedades([nombre_cientifico(equus_caballus), ruido(relincha),
 imagen('caballo.png')])).
frame(rinoceronte_blanco, subclase_de(rhinocerotidae), propiedades([nombre_cientifico(ceratotherium_simum),
  imagen('rino.png')])).
 frame (oso\_pardo, subclase\_de (ursidae), propiedades ([nombre\_cientifico (ursus\_arctos), pelaje (marron), imagen ('pardo.png')])).
 imagen('polar.png')])).
frame(panda, subclase_de(ursidae), propiedades([nombre_cientifico(ailuropoda_melanoleuca),
 pelaje(blanco_y_negro), imagen('panda.png')])).
frame(murcielago, subclase_de(noctilionoidea), propiedades([nombre_cientifico(noctilio_albiventris),
 imagen('dracula.png')])).
frame(vicugna, subclase_de(camelidos), propiedades([nombre_cientifico(vicugna_vicugna),
imagen('vicugna_vicugna.png')])).
  imagen('quanaco.png')])).
  frame(llama, subclase_de(camelidos), propiedades([nombre_cientifico(lama_pacos), imagen('llama.png')])).
 imagen('quagua.png')])).
frame(lobo_rojo, subclase_de(canidae), propiedades([nombre_cientifico(canis_rufus),
imagen('lobo_rojo.png')])).
 frame(babirusa, subclase_de(suidae), propiedades([nombre_cientifico(babyrousa_babyrussa),
imagen('babirusa.png')])).
frame(jabali, subclase_de(suidae), propiedades([nombre_cientifico(pumba_o_sus_scrofa),
imagen('leopardo.png')])).
frame(gato_montes, subclase_de(felidae), propiedades([nombre_cientifico(leopardus_geoffroyi), es(chiquito),
imagen('montes.png')])).
 (que_estA):-((tinstantCta(A),estCtads_A));
(subclase(X),subc(X,(lase))),Clase\=objeto,write('Es '),writeln(Clase),fail.
es(Clase,0bj):- frame(Obj,instancia_de(Clase),_).
es(Clase,0bj):- frame(Obj,instancia_de(Clasep),_),subc(Clasep,Clase).
subc(Cl,C2):- frame(Cl,subclase_de(C2),_).
subc(Cl,C2):- frame(Cl,subclase_de(C2),_),subc(C3,C2).
 subclase(X):-frame(X,subclase_de(_),_).
instancia(X):-frame(X,instancia_de(_),_).
 propiedadesc(X):=frame(X,subclase_de(Y),propiedades(Z)),imprime(Z),propiedadesc(Y).
propiedadesc(X):=frame(X,instancia_de(Y),propiedades(Z)),imprime(Z),propiedadesc(Y).
props(X):=(instancia(X),propiedadesi(X));(subclase(X),propiedadesc(X)).
imprime([]]:=!.
imprime([H|T]):=writeln(H),imprime(T).
```

Iniciando con la descripción de nuestro código en Python, comenzamos estableciendo todos los Labels o elementos de nuestra interfaz tales como el texto, nombre del equipo, los animales disponibles en nuestro programa, etc.

Para poder distribuirlos adecuadamente utilizamos el métido .grid(), propio de la biblioteca Tkinter.

A continuación, tenemos a la función principal que realiza la consulta de nuestro programa prolog a partir del texto ingresado por el usuario.

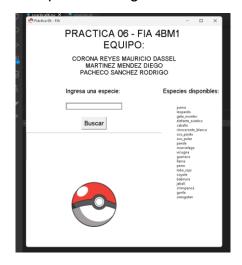
Para esta función, colocamos una caja de texto del tipo textinput, donde el usuario escribe el nombre del animal que quiere conocer, posteriormente creamos un archivo de texto "input.txt" donde escribiremos el texto "about" y el nombre del animal a buscar, esto para poder leer este mismo archivo de texto e insertar este texto para realizar la consulta en nuestro programa de prolog, dicha consulta se imprimirá en otro archivo de texto llamado "output.txt". Finalmente, todo el texto que se haya escrito en nuestro archivo "output.txt" será leído e insertado en una caja de texto para poder mostrarlo en nuestro programa de Python.

Para mostrar la imagen del animal en cuestión, recorreremos el texto proveniente de nuestro "output.txt" hasta encontrar el atributo de "imagen" donde leeremos la ruta de la imagen y la insertaremos en el label correspondiente.

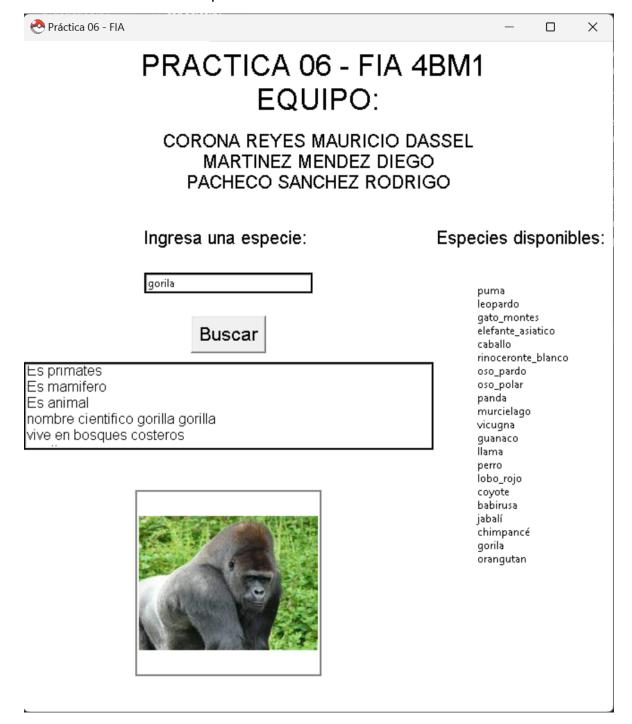
```
def buscar():
    specie = especialinput.get()
    with open("input.txt", "w") as file:
        flie-write(f"about((species)).")
    os.system(f"subjl especies.plt (input.txt > output.txt")
    result = ""
    with open("output.txt", "r") as file:
        result = file.readines()
        cosa = ""
        for x in range(0, len(result)):
            cosa = cosa = "\n" *str(result(x)).replace("\n", "").replace(", "").replace(", "").replace(", "").replace(", "").replace("imagen", "").replace(", " ").replace(", " ").repla
```

Ejecución del programa:

Al ejecutar nuestro programa aparece la siguiente ventana:

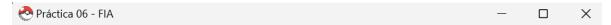


Podemos ver que para facilitar la búsqueda contamos en la parte derecha con un listado de los animales disponibles a consultar.



Al escribir el nombre de algún animal en el listado y dar click en el botón buscar, podemos ver como la información de dicho animal, así como una imagen ilustrativa al mismo, es desplegada en nuestro programa.

Incluso, al intentar buscar un animal que no se encuentre en el listado, aparece un mensaje de error que indica lo siguiente:



PRACTICA 06 - FIA 4BM1 EQUIPO:

CORONA REYES MAURICIO DASSEL MARTINEZ MENDEZ DIEGO PACHECO SANCHEZ RODRIGO

Ingresa una especie:
unicornio
Buscar
Escriba un animal del listado!



Especies disponibles:

puma leopardo gato_montes elefante_asiatico caballo rinoceronte_blanco oso_pardo oso_polar panda murcielago vicugna guanaco llama perro lobo_rojo coyote babirusa jabalí

chimpancé gorila orangutan

.

Conclusiones:

Como pudimos aprender, los sistemas expertos se tratan de programas bastante útiles para la resolución de problemas, capaces de razonar por medio de relaciones lógicas y fundamentados en el conocimiento que les proveamos. Al investigar un poco más aprendimos que los sistemas expertos recientemente diseñados buscan crear tomas de decisiones más eficaces y reducir el tiempo de decisión, actualmente aún no se considera que superen a expertos humanos en la toma de decisiones, pero se espera que estos sistemas puedan aportar técnicas de inducción y razonamiento complementarias a las propuestas por expertos humanos y así ayudar a una toma de decisiones más eficaz colaborando tanto la máquina como el humano.

Referencias

Harmon, N. P., & King, D. R. (1985). *Expert Systems: Artificial Intelligence in Business*. John Wiley & Sons.

Clocksin, W. F., & Mellish, C. S. (1990). *Programmieren in Prolog* (H. Sterner, Trad.; 1990a ed.). Springer.