



**INSTITUTO POLITECNICO  
NACIONAL**  
ESCUELA SUPERIOR DE COMPUTO



# **Instituto Politécnico Nacional**

## **Escuela Superior de Cómputo**

### **Fundamentos de Inteligencia Artificial**

## **Reporte de práctica 6**

**PROFESOR: Macario Hernandez Cruz**

**GRUPO: 4BM1**

### **INTEGRANTES:**

Carmona Serrano Ian Carlo

Mendez Lopez Luz Fernanda

Rosales Benhumea Aldo

## Práctica 5

### Lenguaje de predicados

#### RESUMEN

En esta práctica desarrollamos un programa que se conecta a una base de datos de Prolog y tiene una interfaz en Java. El objetivo del programa es permitir al usuario obtener información sobre los animales de la base de datos. Al hacer una pregunta sobre un animal específico, el programa devuelve todas sus características almacenadas en la base de datos, además contamos con una opción en la que se pueden visualizar todos los animales registrados.

La base de datos de Prolog utilizada en esta práctica contiene información sobre una variedad de animales.

El programa utiliza la lógica de programación en Prolog para buscar la información en la base de datos y devolverla a la interfaz de usuario en Java.

Prolog es su capacidad para trabajar con declaraciones lógicas y deducir conclusiones.

En esta práctica se busca implementar una herramienta que permita visualizarla foto y la información más relevante que se tiene registrada de un animal.



#### INTRODUCCIÓN

Prolog es un lenguaje de programación basado en la lógica formal. Fue creado en la Universidad de Marsella por Alain Colmerauer y Philippe Roussel a finales de los años 70, fue creado con el objetivo de facilitar la resolución de problemas complejos mediante la lógica. Desde entonces, ha sido utilizado en diversas áreas entre las más importantes la informática e inteligencia artificial.

En la actualidad, existen diversas implementaciones de Prolog, como SWI-Prolog, GNU Prolog o XSB, entre otros.

Una de las principales características de



## FUNCIONES

### FRAMES:

Los frames son los que nos permiten definir a los animales, es nuestra base para poder deducir sus características y poder clasificarlos, los predicados están hechos en prolog lo que buscaremos es pasar estas reglas a python para poder arrojar la salida solicitada.

```
%reino
frame(animal, subclase_de(objeto), propiedades([puede(sentir), puede(respirar)])).

%tipo
frame(mamifero, subclase_de(animal), propiedades([puede(mamar), respira(aire)])).

%Orden
frame(artiodactilo, subclase_de(mamifero, propiedades([tiene(pesugnas_pares), comen(plantas)]))).
frame(carnivora, subclase_de(mamifero), propiedades([comen(carne)])).
frame(primates, subclase_de(mamifero), propiedades([tiene(cerebro_desarrollado)])).
frame(proboscidea, subclase_de(mamifero), propiedades([es(grande)])).
frame(perissodactyla, subclase_de(mamifero), propiedades([tiene(pesugnas_impares)])).
frame(chiroptera, subclase_de(mamifero), propiedades([tien(alas), es(roedor)])).

%Familia
frame(camelidos, subclase_de(artiodactilo), propiedades([familia_de(camellos)])).
frame(canidae, subclase_de(carnivora), propiedades([puede(comer_vegetales)])).
frame(suidae, subclase_de(artiodactilo), propiedades([son(inteligentes)])).
frame(hominidae, subclase_de(primates), propiedades([son(grandes_simios)])).
frame(felidae, subclase_de(carnivora), propiedades([son(felinos)])).
frame(elephantidae, subclase_de(proboscidea), propiedades([son(elefantes)])).
frame(equidae, subclase_de(perissodactyla), propiedades([son(compatibles)])).
frame(rhinocerotidae, subclase_de(perissodactyla), propiedades([tien(cuerno)])).
frame(ursidae, subclase_de(carnivora), propiedades([son(osos)])).
frame(noctilionoidea, subclase_de(chiroptera), propiedades([es(pescador)])).

%Ejemplares
frame(puma_concolor, subclase_de(felidae), propiedades([nombre_comun(puma), es(pequenia), emite(maullidos), imagen('puma.jpg')])).
frame(panthera_pardus, subclase_de(felidae), propiedades([nombre_comun(leopardo), es(rapido), imagen('leopardo.jpg')])).
frame(leopardus_geoffroyi, subclase_de(felidae), propiedades([nombre_comun(gato_montes), es(chiquito), imagen('montes.jpg')])).
frame(elephas_maximus, subclase_de(elephantidae), propiedades([nombre_comun(elefante_asiatico), imagen('elefante.jpg')])).
frame(equus_caballus, subclase_de(equidae), propiedades([nombre_comun(caballo), ruido(relincha), imagen('caballo.jpg')])).
frame(ceratotherium_simum, subclase_de(rhinocerotidae), propiedades([nombre_comun(rinoceronte_blanco), imagen('rino.jpg')])).
frame(ursus_arctos, subclase_de(ursidae), propiedades([nombre_comun(oso_pardo), pelaje(marron), imagen('pardo.jpg')])).
frame(ursus_maritimus, subclase_de(ursidae), propiedades([nombre_comun(oso_polar), pelaje(blanco), imagen('polar.jpg')])).
frame(ailuropoda_melanoleuca, subclase_de(ursidae), propiedades([nombre_comun(oso_panda), pelaje(blanco_y_negro), imagen('panda.jpg')])).
frame(noctilio_albiventris, subclase_de(noctilionoidea), propiedades([nombre_comun(murcielago_bulldog), imagen('murcielago.jpg')])).
|
que_es(X):-((instancia(X),es(Clase,X));
```



```
(subclass(X),subc(X,Clase)),Clase\=objeto,write('es '),writeln(Clase),fail.
es(Clase,Obj):- frame(Obj,instancia_de(Clase),_).
es(Clase,Obj):- frame(Obj,instancia_de(Clasep),_),subc(Clasep,Clase).

subc(C1,C2):- frame(C1,subclass_de(C2),_).
subc(C1,C2):- frame(C1,subclass_de(C3),_),subc(C3,C2).
subclass(X):-frame(X,subclass_de(_),_).

instancia(X):-frame(X,instancia_de(_),_).

propiedades(objeto):-!.
propiedades(X):-frame(X,subclass_de(Y),propiedades(Z)),imprime(Z),propiedades(Y).
propiedades(X):-frame(X,instancia_de(Y),propiedades(Z)),imprime(Z),propiedades(Y).
props(X):-(instancia(X),propiedades(X));(subclass(X),propiedades(X)).

imprime([]):-!.
imprime([H|T]):-writeln(H),imprime(T).
about(X):-que_es(X);props(X).
```

### LIBRERÍAS DE PYTHON UTILIZADAS:

tkinter -> interfaz gráfica.

os -> llamar al sistema.

pyswip -> conexión a prolog.

PIL -> imágenes.

json -> formato de texto.

```
from tkinter import *
import os
from pyswip import Prolog
from PIL import Image,ImageTk
from json import *
```

Center\_window: Nos va a servir para poder centrar la interfaz gráfica en el ordenador que se esté utilizando

```
def center_window(window):
    window.update_idletasks()
    width = window.winfo_width()
    height = window.winfo_height()
    x = (window.winfo_screenwidth() // 2) - (width // 2)
    y = (window.winfo_screenheight() // 2) - (height // 2)
    window.geometry('{}x{}+{}+{}'.format(width, height, x, y))
```

Consultar: Nos servirá para redireccionar a la funciones siguientes dependiendo lo que extraiga de “elección”

```
def consulta():
    global eleccion,root
    if eleccion.get()==1:
        consult_specie()
    if eleccion.get()==2:
        list_species()
```

Consult Specie: Esta función lo que realizara sera conectarse al prolog le pasaremos la ruta donde esta el archivo que queremos leer y también extraemos el input que introdujo el usuario, después abriremos un archivo de texto el cual guardará lo que introdujo el usuario, después el sistema ejecutará un comando el cual del archivo prolog seleccionado va a leer el input y la respuesta lo va a regresar a un output el cual lo leeremos y ahí se encontrarán todas las características del animal introducido después insertamos la respuesta en la interfaz gráfica junto con su imagen la cual será buscada en una carpeta con el nombre del animal.

```
def consult_specie():
    global especieInput, textResult, labelImg, image2
    prolog = Prolog()
    prolog.consult('./animalia.pl')
    specie = especieInput.get()

    with open("input.txt", "w") as file:
        file.write(f"about({specie}).")

    os.system("swipl animalia.pl < input.txt > output.txt")

    with open("output.txt", "r") as file:
        content = file.read()

    textResult.delete("1.0", "end")
    textResult.insert(END, content)

    for line in content.splitlines():
        if "imagen" in line:
            src = line.split("(")[1].split(")")[0]

            image2 = Image.open(f"img/{src}")
            image2 = image2.resize((100, 100), Image.Resampling.LANCZOS)
            imgTk = ImageTk.PhotoImage(image2)

            labelImg2 = Label(frame, image=imgTk)
            labelImg2.place(x=40, y=20)
            labelImg2.image = imgTk
```

List species: esta función listará todas las especies que están disponibles para consultar y las imprimirá en la interfaz gráfica con un formato para que puedan ser más legibles.

```
def list_species():
    global textResult

    animals = [
        "puma_concolor",
        "panthera_pardus",
        "leopardus_geoffroyi",
        "elephas_maximus",
        "equus_caballus",
        "ceratotherium_simum",
        "ursus_arctos",
        "ursus_maritimus",
        "ailuropoda_melanoleuca",
        "noctilio_albiventris"
    ]

    textResult.delete("1.0", "end")
    formatted_animals = json.dumps(animals, indent=4)
    textResult.insert(END, formatted_animals)
```

Esto es el resto de código digamos que es la función principal, lo que hace esta parte de código es crear la interfaz gráfica y configurarla a nuestro gusto, podemos también observar que hay unas funciones las cuales nos permiten redimensionar las imágenes para que siempre tengan el mismo tamaño y al momento de ejecutar el botón “consultar” le mandamos la función de consulta la cual como ya habíamos dicho va a elegir la función correspondiente a realizar.

```
root = Tk()
root.resizable(False,False)
frame = Frame(root)
frame.pack(fill="both",expand=True)
frame.config(width=450,height=350, bg='blue' )
center_window(root);

textResult = Text(width=30,height=20,bg='lightgreen')
textResult.place(x=200,y=10)

image=Image.open("akinator.jpg")
image=image.resize((100,100),Image.Resampling.LANCZOS)
img=ImageTk.PhotoImage(image)
lbl_img=Label(frame,image=img)
lbl_img.place(x=40,y=20)

especieInput = Entry(width=21,bg='lightgreen')
especieInput.place(x=40,y=180)

specieLabel = Label(text="Que especie desea buscar:", bg='blue', fg='white', font=("Trebuchet MS", 10))
specieLabel.place(x=20,y=150)

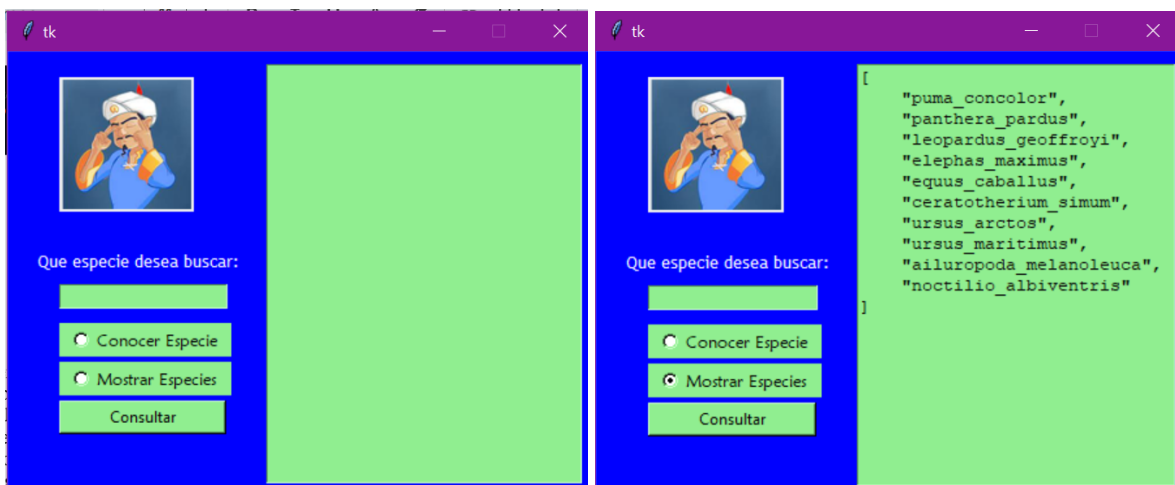
eleccion = IntVar()
btnSpecie = Radiobutton(text="Conocer Especie",width=15,bg='lightgreen',font=("Maiandra GD",10),value=1,variable=eleccion)
btnSpecie.place(x=40,y=210)

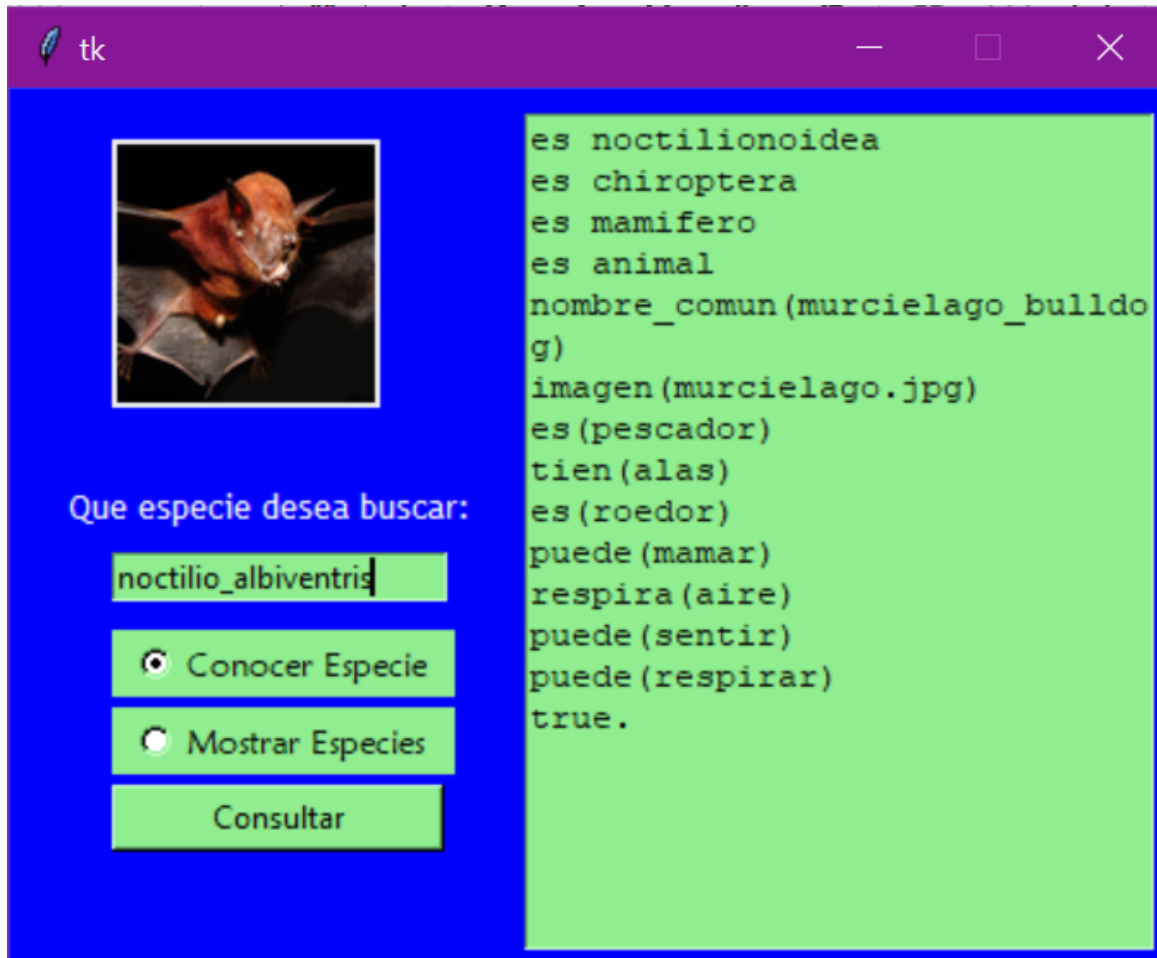
btnListAll = Radiobutton(text="Mostrar Especies",width=15,bg='lightgreen',font=("Maiandra GD",10),value=2,variable=eleccion)
btnListAll.place(x=40,y=240)

consultar = Button(text="Consultar",width=17,bg='lightgreen',command=consulta)
consultar.place(x=40,y=270)

root.mainloop()
```

## RESULTADO





## CONCLUSIÓN

En esta práctica implementamos los conceptos aprendidos en las clases de fundamentos de inteligencia artificial para realizar las consultas de los animales solicitados, además con la ayuda de las primeras prácticas realizamos una interfaz para comunicar a los usuarios con la base de datos, la mayor dificultad que encontramos fue al conectar la base de datos, ya que intentamos con el programa que teníamos instalado de Prolog y no logramos una conexión, por lo que probamos con ocupar otra versión de prolog y logramos una comunicación exitosa.