

EXPLORANDO LA BIODIVERSIDAD EN CUNDINAMARCA Y BOYACA

FASE II

AÑO 2024

FASE II PROYECTO FINAL: EXPLORANDO LA BIODIVERSIDAD EN CUNDINAMARCA Y BOYACA

Introducción: En esta fase, nos enfocaremos en utilizar Python para agregar funcionalidades más avanzadas a nuestra aplicación sobre la biodiversidad en Colombia. Nos proponemos **recopilar y procesar datos de diversas fuentes** para ofrecer información detallada y útil sobre la biodiversidad. Además, integraremos estas funcionalidades con la base visual establecida en la Fase I.

OBJETIVO GENERAL:

Desarrollar una aplicación web interactiva que permita a los usuarios explorar y aprender sobre la biodiversidad en Colombia, utilizando conceptos aprendidos en HTML, CSS, JavaScript, Python, Pandas y NumPy.

OBJETIVO ESPECIFICO:

Implementación de Funcionalidades de Recopilación y Procesamiento de Datos

TAREAS ESPECIFICAS:

1. Recopilación de Datos:

- Utilizar Python para acceder y recopilar datos relevantes sobre la biodiversidad en Colombia. Pueden ser datos climáticos, información sobre especies, o cualquier conjunto de datos pertinente.
- Utilizar bibliotecas como **requests** o **BeautifulSoup** para la extracción de datos web, si es necesario.

2. Procesamiento de Datos:

- Implementar funciones en Python para procesar y organizar los datos recopilados.
- Utilizar estructuras de datos como listas, diccionarios

3. Integración con la Interfaz Web:

- Conectar las funcionalidades de Python con la interfaz web creada en la Fase I.
- Mostrar dinámicamente información procesada en la página web utilizando Flask u otro marco web si es necesario.

ENTREGABLE:

Una aplicación web funcional que incorpora datos recopilados y procesados en Python, ofreciendo a los usuarios información detallada y actualizada sobre la biodiversidad en Cundinamarca y Boyacá.

1. Recopilación de Datos:

- Utilizar Python para recopilar datos relevantes sobre la biodiversidad en Cundinamarca y Boyacá.

- Puedes considerar el uso de librerías como **requests** para obtener datos de API relacionadas con biodiversidad.

2. Análisis Básico:

- Aplicar conceptos de Python para realizar un análisis básico de los datos recopilados.
- Puedes utilizar estructuras de datos como listas y diccionarios para organizar y manipular la información.



```

1  ▾ datosAves= [
2  ▾      {
3          "nombreComun": "Colibrí",
4  ▾      "descripcion": {
5          "tamaño": "9 cm",
6          "color": "Verde y Rojo",
7          "caracteristica": "Rapido",
8          "comportamiento": "Se alimenta de néctar."
9      },
10 ▾      "multimedia": {
11          "fotos": ["ruta/foto1.jpg", "ruta/foto2.jpg"],
12          "sonidos": ["ruta/sonido1.jpg"]
13      },
14 ▾      "observaciones": [
15          {"fecha": "2024-05-02", "lugar": "Boyacá", "avistamientos": 5},
16          {"fecha": "2024-04-10", "lugar": "Cundinamarca", "avistamientos": 10}
17      ]
18      }
19 ]

```

```

22 def mostrarAves():
23     print("Mostrar aves")
24     print(datosAves)

```

```

27 def agregarAve():
28     print("Agregar ave")
29     nuevaAve= {}
30
31     nombre= input("Ingrese el nombre común: ")
32
33     nuevaAve["nombreComun"]= nombre
34
35     descripcion= {}
36     descripcion["tamaño"]= input("Ingrese el tamaño: ")
37     descripcion["color"]= input("Ingrese el color: ")
38     descripcion["caracteristica"]= input("Ingrese el caracteristica: ")
39     descripcion["comportamiento"]= input("Ingrese el comportamiento: ")
40
41     nuevaAve["descripcion"]= descripcion
42
43     multimedia= {}
44     multimedia["fotos"]= input("Ingrese las rutas de las fotos, separadas por comas (ruta/foto1.jpg, ruta/foto2.jpg): ").split(",")
45     multimedia["sonidos"]= input("Ingrese las rutas de los sonidos, separadas por comas (ruta/sonido1.jpg, ruta/sonido2.jpg): ").split(",")
46
47     nuevaAve["multimedia"]= multimedia

```

```

48
49 observaciones= []
50 while True:
51     res= input("¿Desea agregar una observación?: \n1. si \n2. no ")
52     if res == "2":
53         break
54
55     observacion= {}
56     observacion["fecha"]= input("Ingrese la fecha de la observación (YYYY-MM-DD): ")
57     observacion["lugar"]= input("Ingrese el lugar de la observación: ")
58     observacion["avistamientos"]= int(input("Ingrese el numero de avistamientos: "))
59
60     observaciones.append(observacion)
61
62 nuevaAve["observaciones"]= observaciones
63
64 datosAves.append(nuevaAve)
65
66 print("Ave agregada.")

```

```

69 def actualizarAve():
70     print("Actualizar aves")
71     nombre= input("Ingrese el nombre común: ")
72     aveEncontrada= False
73
74     for ave in datosAves:
75         if ave["nombreComun"] == nombre:
76             aveEncontrada= True
77             print("Seleciones el campo que desea actualizar: ")
78             print("1. Descripción")
79             print("2. multimedia")
80             print("3. observaciones")
81
82             opcion= input("Seleccione una opción: ")
83
84             if opcion == "1":
85                 descripcion= {}
86                 descripcion["tamaño"]= input("Ingrese el nuevo tamaño: ")
87                 descripcion["color"]= input("Ingrese el nuevo color: ")
88                 descripcion["característica"]= input("Ingrese la nueva característica: ")
89                 descripcion["comportamiento"]= input("Ingrese el nuevo comportamiento: ")
90                 ave["descripcion"]= descripcion
91
92             elif opcion == "2":
93                 multimedia= {}
94                 multimedia["fotos"]= input("Ingrese las nuevas rutas de las fotos, separadas por comas (ruta/foto1.jpg, ruta/foto2.jpg): ").split(",")
95                 multimedia["sonidos"]= input("Ingrese las nuevas rutas de los sonidos, separadas por comas (ruta/sonido1.jpg, ruta/sonido2.jpg): ").split(",")
96                 ave["multimedia"]= multimedia

```

```

98         elif opcion == "3":
99             observaciones= []
100             while True:
101                 res= input("¿Desea agregar una observación?: \n1. si \n2. no ")
102                 if res == "2":
103                     break
104
105                 observacion= {}
106                 observacion["fecha"]= input("Ingrese la nueva fecha de la observación (YYYY-MM-DD): ")
107                 observacion["lugar"]= input("Ingrese el nuevo lugar de la observación: ")
108                 observacion["avistamientos"]= int(input("Ingrese el nuevo numero de avistamientos: "))
109
110                 observaciones.append(observacion)
111
112             ave["observaciones"]= observaciones
113
114             print("Ave actualizada.")
115
116 if not aveEncontrada:
117     print("Nombre comun no encontrado.")

```

```

120 def eliminarAve():
121     print("Eliminar aves")
122     nombre= input("Ingrese el nombre comun que desea eliminar: ")
123     longitudAnterior= len(datosAves)
124     datosAves[:] = [ ave for ave in datosAves if ave["nombreComun"] != nombre ]
125     if len(datosAves) < longitudAnterior:
126         print("Ave eliminada.")
127     else:
128         print("Nombre comun no encontrado.")

```

```
131 def menu():
132     while True:
133         print("\nMenú de Gestion de Aves:")
134         print("1. Ver todas las especies")
135         print("2. Agregar nueva especie")
136         print("3. Actualizar datos de una especie")
137         print("4. Eliminar una especie")
138         print("5. Salir")
139
140         opcion= input("Seleccione una opción: ")
141
142         if opcion == "1":
143             mostrarAves()
144         elif opcion == "2":
145             agregarAve()
146         elif opcion == "3":
147             actualizarAve()
148         elif opcion == "4":
149             eliminarAve()
150         elif opcion == "5":
151             print("Salir")
152             break
153         else:
154             print("Opción no valida.")
155
156     menu()
```