

**LOCALIZACIÓN DE ESPECIES DE ÁRBOLES
ENDÉMICAS EN EL MUNICIPIO DE TUTA**

Estudiante

Milton Fredy Figueredo Medina

TALENTO TECH

UNIVERSIDAD SERGIO ARBOLEDA

MinTIC

Profesor

Rodolfo Ortega

20 de Septiembre del 2024

Colombia

CONTENIDO

INTRODUCCIÓN.....	3
1.Localización De Especies De Árboles Endémicas En El Municipio De Tuta.....	4
2. Situaciones o problemas.....	5
3. Necesidad de Negocio.....	4
4. Objetivos.....	4
5. Entregable.....	4
6. Levantamiento de la Información.....	5
6.1. Entrevistas con Expertos.....	5
6. 2. Análisis de Documentación.....	5
6. 3. Workshops con Stakeholders.....	6
7. Código CRUD Python Proyecto De Creación 3.....	7
9. CONCLUSIONES.....	15

INTRODUCCIÓN

Este proyecto tiene como objetivo crear conciencia sobre la importancia de preservar los árboles endémicos dentro del municipio de Tuta en el departamento de Boyacá.

1. Localización De Especies De Árboles Endémicas En El Municipio De Tuta

2. Situaciones o problemas

En el Municipio de Tuta en Boyacá existe desconocimiento generalizado de la comunidad sobre las especies endémicas de árboles que hay en Boyacá, estas especies tienen características especiales, son resistentes al clima frío, al exceso de lluvias y a las temporadas de verano.

Además estos árboles tienen microsistemas que mantienen en equilibrio el ecosistemas de los entornos.

El desconocimiento y el avance descontrolado en construcciones en los entornos rurales está generando una tala de estos árboles endémicos, que pueden mantener el ecosistema en equilibrio y así tratar de contrarrestar en el verano la falta de agua.

3. Necesidad de Negocio

En una página web online 24/7 las personas pueden saber cuáles árboles y en cuáles circunstancias específicas pueden ser talados.

4. Objetivos

Generar conciencia del cuidado de los árboles con planes de salidas como caminatas o salidas ciclistas, para generar en las personas impactos para realizar cambios.

5. Entregable

Título: Especies De Árboles Endémicas En El Municipio De Tuta.

Título comercial: Biodiversidad de Árboles en Tuta.

Requerimientos del sistema: la página web debe estar disponible 24/7, debe poseer dominio y hosting en versión gratuitas para realizar la validación del proyecto con un prototipo

funcional en alta calidad. Donde los usuario puedan revisar la información por medio de internet o datos.

6. Levantamiento de la Información:

Aun estoy en la búsqueda del levantamiento de la información, ya he visto algunos puntos importantes a seguir.

6.1 Entrevistas con Expertos:

- Propósito: Recolectar información detallada sobre las especies de árboles endémicos de Boyacá y sus necesidades específicas para diseñar funcionalidades clave del sitio web.
- Participantes: Botánicos, Ingenieros Ambientales, y expertos en biodiversidad.
- Resultado Esperado: Obtención de datos precisos sobre las características de las especies, su distribución geográfica, y los desafíos de conservación.

6. 2. Análisis de Documentación:

- Propósito: Revisar leyes, regulaciones, y políticas locales sobre la tala de árboles y la protección de especies endémicas para asegurar que el sitio web cumpla con las normativas legales.
- Fuente de Información: Documentos gubernamentales, estudios previos, y reportes ambientales.
- Resultado Esperado: Identificación de restricciones legales y requisitos para el desarrollo de contenido informativo y educativo en la plataforma.

6. 3. Workshops con Stakeholders:

- **Propósito:** Organizar talleres colaborativos con los stakeholders para definir los requerimientos del sistema, priorizar funcionalidades y crear prototipos iniciales.
- **Participantes:** Representantes de la Alcaldía municipal, miembros de la comunidad, y profesionales como Ingenieros Civiles y Maestros de Obra.
- **Resultado Esperado:** Clarificación del alcance del proyecto y alineación de las expectativas de todos los involucrados en el desarrollo del sitio web.

7. Código CRUD Python Proyecto De Creación 3

```

proyecto > backend > main_Arboles_03.py > ...
1  import pandas as pd
2  import matplotlib.pyplot as plt
3  import numpy as np
4  import random
5
6  datos_arboles=[
7      {
8          "id": id,
9          "nombre": "Aliso",
10         "descripcion": {
11             "tamano": 30,
12             "color": "Verde oscuro.",
13             "caracteristica": "Fuerte.",
14             "comportamiento": "Hábitat natural en los lugares húmedos."
15         },
16         "multimedia": {
17             "fotos": ["foto1.jpg", "foto2.jpg"],
18             "videos": ["video1.mp4"]
19         },
20         "observaciones": [
21             {"fecha": "2022-04-25", "lugar": "Boyacá - Tuta", "avistamientos": 400},
22             {"fecha": "2023-09-10", "lugar": "Boyacá - Tuta", "avistamientos": 200}
23         ]
24     }
25 ]
26
27
28
29 def mostrar_arboles():
30     datos=[]
31
32     for arbol in datos_arboles:
33
34
35         observaciones_fecha=[observacion["fecha"] for observacion in arbol["observaciones"]]
36         observaciones_lugar=[observacion["lugar"] for observacion in arbol["observaciones"]]
37         observaciones_avistamientos=[observacion["avistamientos"] for observacion in arbol["observaciones"]]
38
39         datos.append({
40             "ID": arbol["id"],
41             "Nombre": arbol["nombre"],
42             "Tamaño": arbol["descripcion"]["tamano"],
43             "Color": arbol["descripcion"]["color"],
44             "Caracteristica": arbol["descripcion"]["caracteristica"],
45             "Comportamiento": arbol["descripcion"]["comportamiento"],
46             "Fotos": ", ".join(arbol["multimedia"]["fotos"]),
47             "Videos": ", ".join(arbol["multimedia"]["videos"]),
48             "Fecha": ", ".join(map(str, observaciones_fecha)),
49             "Lugar": ", ".join(map(str, observaciones_lugar)),
50             "Avistamientos": ", ".join(map(str, observaciones_avistamientos))
51         })
52
53     df=pd.DataFrame(datos)
54     print(df)
55

```

```

proyecto > backend > main_Arboles_03.py > ...
56 def agregar_arbol():
57     global id
58     id+=1
59
60     nombre=input("Ingrese el nombre: ")
61
62     descripcion={}
63     descripcion["tamano"]=int(input("Ingrese el tamaño del Árbol: "))
64     descripcion["color"]=input("Ingrese el color del Árbol: ")
65     descripcion["caracteristica"]=input("Ingrese la característica del Árbol: ")
66     descripcion["comportamiento"]=input("Ingrese la comportamiento del Árbol: ")
67
68     multimedia={}
69     multimedia["fotos"]=input("Ingrese los nombres de las fotos: (foto1.jpg , foto2.jpg): ").split(",") # ["ruta/foto1.jpg , ruta/foto2.jpg"]
70     multimedia["videos"]=input("Ingrese los nombres de los vídeos: (videos1.mp3 , videos2.mp3): ").split(",") # ["ruta/foto1.jpg , ruta/foto2.jpg"]
71
72     observaciones=[]
73     while True:
74
75         observacion={}
76         observacion["fecha"]=input("Ingrese la fecha de la observación (YYY-MM-DD) o 'fin' para terminar: ")
77
78         if observacion["fecha"].lower() == "fin":
79             break
80
81         observacion["lugar"]=input("Ingrese el lugar de la observación: ")
82         observacion["avistamientos"]=input("Ingrese el números de avistamientos: ")
83
84         observaciones.append(observacion)
85
86     datos_arboles.append({
87         "id": id,
88         "nombre": nombre,
89         "descripcion": descripcion,
90         "multimedia": multimedia,
91         "observaciones": observaciones
92     })
93
94

```



```

proyecto > backend > main_Arboles_03.py > ...
91     OBSERVACIONES : OBSERVACIONES
92
93     })
94
95 def actualizar_arbol():
96     id_arbol= int(input("Ingrese el ID del Árbol que desea actualizar: "))
97     arbol_encontrado= False
98
99     for arbol in datos_arboles:
100         if id_arbol == arbol["id"]:
101             arbol_encontrado=True
102             print("Seleccione la opción que desea actualizar: ")
103             print("1. Descripción")
104             print("1. Multimedia")
105             print("1. Observaciones")
106
107             opcion=input()
108
109             if opcion == "1":
110                 descripcion={}
111                 descripcion["tamano"]=int(input("Ingrese el nuevo tamaño del Árbol: "))
112                 descripcion["color"]=input("Ingrese el nuevo color del Árbol: ")
113                 descripcion["caracteristica"]=input("Ingrese las nuevas característica del Árbol: ")
114                 descripcion["comportamiento"]=input("Ingrese el nuevo comportamiento del Árbol: ")
115                 arbol["descripcion"]=descripcion
116
117             elif opcion == "2":
118                 multimedia={}
119                 multimedia["fotos"]=input("Ingrese los nombres de las nuevas fotos: (foto1.jpg , foto2.jpg): ").split(",") # ["ruta/foto1.jpg , ruta/foto2.jpg"]
120                 multimedia["videos"]=input("Ingrese los nombres de los nuevos videos: videos1.mp3 ,videos2.mp3): ").split(",") # ["ruta/foto1.jpg , ruta/foto2.jpg"]
121                 arbol["multimedia"]=multimedia
122
123             elif opcion == "3":
124                 observaciones=[]
125                 while True:
126                     observacion={}
127                     observacion["fecha"]=input("Ingrese la nueva fecha de la observación (YYY-MM-DD):")
128
129                     if observacion["fecha"].lower() == "fin":
130                         break
131
132                     observacion["lugar"]=input("Ingrese el nuevo lugar de la observación: ")
133                     observacion["avistamientos"]=input("Ingrese el números de nuevos avistamientos: ")
134
135                     observaciones.append(observacion)
136
137                     arbol["observaciones"]=observaciones
138
139             print("Ave agregada")
140
141
142 if not arbol_encontrado:
143     print("Nombre no encontrado.")
144

```

```

145 def eliminar_arbol():
146     id_arbol= int(input("Ingrese el ID del ave que desea actualizar: "))
147     longitudAnterior = len(datos_arboles)
148
149     datos_arboles[:]=[arbol for arbol in datos_arboles if id_arbol != arbol["id"]]
150
151     if longitudAnterior > len(datos_arboles):
152         print("Árbol eliminada")
153     else:
154         print("ID no encontrado.")
155
156 def analisis_datos():
157     observaciones=[]
158
159     for arbol in datos_arboles:
160         for observacion in arbol["observaciones"]:
161             observaciones.append({
162                 "Nombre": arbol["nombre"],
163                 "Fecha": observacion["fecha"],
164                 "Lugar": observacion["lugar"],
165                 "Avistamientos": observacion["avistamientos"]
166             })
167
168     df=pd.DataFrame(observaciones)
169     print(df)
170
171     if df.empty:
172         print("No hay datos disponibles para el análisis.")
173         return
174
175     print("\nAnálisis de datos.")
176     print("\nEstadística descriptivas antes de la limpieza.")
177     print(df.describe(include="all"))
178
179     df=df.drop_duplicates()
180     print(df)
181
182     df=df.dropna(subset=["Nombre", "Fecha"])

```

proyecto > backend > main_Arboles_03.py > ...

```

179     df=df.drop_duplicates()
180     print(df)
181
182     df=df.dropna(subset=["Nombre","Fecha"])
183     print(df)
184
185     df["Avistamientos"]=df["Avistamientos"].fillna(df["Avistamientos"].median())
186     print(df)
187
188     df["Fecha"]=pd.to_datetime(df["Fecha"])
189     print(df)
190
191     df=df[ ( df["Avistamientos"]>0 ) & ( df["Avistamientos"] <= 100 ) ]
192     print(df)
193
194     print("\nEstadística descriptivas despues de la limpieza.")
195     print(df.describe(include="all"))
196
197     # Establecer la fecha como índice DF
198     df.set_index("Fecha", inplace=True)
199     print(df)
200
201     print("\nTendencia a lo largo del tiempo por mes.")
202     tendencia=df.resample("ME").sum(numeric_only=True)
203     print(tendencia)
204
205
206     print("\nDistribución de avistamientos por lugar")
207     distribucion=df.groupby("Lugar").sum(numeric_only=True)
208     print(distribucion)
209
210     print("\nAvistamientos proyecto árbol")
211     avistamientos=df.groupby("Nombre").sum(numeric_only=True)
212     print(avistamientos)
213
214     # grafica
215
216     print("Crear grafica")
217
218     fig= plt.figure(figsize=(14,10))
219     fig.canvas.manager.set_window_title("Análisi de datos arboles")
220
221     plt.subplot(2,2,1)
222     tendencia["Avistamientos"].plot(kind="line", marker="o", color="red")
223     plt.title("Tendencia a lo largo del tiempo por mes")
224     plt.xlabel("Fecha")
225     plt.ylabel("Avistamiento")
226
227     plt.subplot(2,2,2)
228     distribucion["Avistamientos"].plot(kind="bar",color="blue")
229     plt.title("distribucion de avistamientos por lugar")
230     plt.xlabel("Lugar")
231     plt.ylabel("Avistamiento")
232
233     plt.subplot(2,2,3)
234     avistamientos["Avistamientos"].plot(kind="hist", bins=10, color="purple", alpha=0.7)
235     plt.title("Avistamientos promedio por ave")
236     plt.xlabel("Avistamientos")
237     plt.ylabel("Frecuencia")

```

```

proyecto > backend > main_Arboles_03.py > ...
237 plt.ylabel("Frecuencia")
238
239 plt.tight_layout()
240 plt.get_current_fig_manager().window.state("zoomed")
241 plt.show()
242
243 plt.subplot(2,2,4)
244 colores=plt.cm.Paired(np.arange(len(avistamientos)))
245 avistamientos["Avistamientos"].plot(kind="pie", color= colores, startangle=90)
246 plt.title("Avistamientos promedio por árbol")
247 plt.ylabel("Avistamientos")
248
249 def cargar_datos():
250     global id, datos_arboles
251
252     try:
253         df=pd.read_csv("datos_arboles.csv")
254
255         datos_arboles=[]
256         arboles={}
257
258         for _, row in df.iterrows():
259             id_arbol=row["ID"]
260
261             if id_arbol not in arboles:
262                 arboles[id_arbol]= {
263                     "id": id_arbol,
264                     "nombre": row["Nombre"],
265                     "descripcion": {
266                         "tamano": row["Tamaño"],
267                         "color": row["Color"],
268                         "caracteristica": row[ "Caracteristica"],
269                         "comportamiento": row[ "Comportamiento"]
270                     },
271                     "multimedia": {
272                         "fotos": row[ "Fotos"].split(", "),
273                         "videos": row[ "Videos"].split(", ")
274                     },
275                     "observaciones": [ ]
276                 }
277
278                 arboles[id_arbol]["observaciones"].append({
279                     "fecha": row["Fecha"],
280                     "lugar": row["Lugar"],
281                     "avistamientos": row["Avistamientos"]
282                 })
283
284             # print(aves)
285
286             datos_arboles=list(arboles.values())
287             id=max(arboles.keys())
288
289             print(datos_arboles)
290
291             print("Datos cargados desde el archivo.")
292     except FileNotFoundError:
293         print("No se encuentra el archivo. Generando datos")
294         generarDatos(datos_arboles, 5)
295

```

proyecto > backend > main_Arboles_03.py > ...

```

295
296 def guardar_datos():
297     datos=[]
298
299     for arbol in datos_arboles:
300         for observaciones in arbol["observaciones"]:
301             datos.append({
302                 "ID": arbol["id"],
303                 "Nombre": arbol["nombre"],
304                 "Tamaño": arbol["descripcion"]["tamano"],
305                 "Color": arbol["descripcion"]["color"],
306                 "Caracteristica": arbol["descripcion"]["caracteristica"],
307                 "Comportamiento": arbol["descripcion"]["comportamiento"],
308                 "Fotos": "", ".join(arbol["multimedia"]["fotos"]),
309                 "Videos": "", ".join(arbol["multimedia"]["videos"]),
310                 "Fecha": observaciones["fecha"],
311                 "Lugar": observaciones["lugar"],
312                 "Avistamientos": observaciones["avistamientos"]
313             })
314
315     df=pd.DataFrame(datos)
316     df.to_csv("datos_arboles.csv", index=False)
317     print("Datos gurdados en el archivo datos_arboles.csv")
318
319 def generarObservaciones():
320     pass
321
322 def generarDatos():
323     pass
324

```

```

proyecto > backend > main_Arboles_03.py > ...
313         })
314
315     df=pd.DataFrame(datos)
316     df.to_csv("datos_arboles.csv", index=False)
317     print("Datos gurdados en el archivo datos_arboles.csv")
318
319     def generarObservaciones():
320         pass
321
322     def generarDatos():
323         pass
324
325     def menu():
326         cargar_datos()
327
328         while True:
329             print("\n---Menú gestión de Árboles---")
330             print("\n1. Ver todos los Árboles.")
331             print("2. Agregar nuevo Árbol.")
332             print("3. Actualizar datos de un Árbol.")
333             print("4. Eliminar un Árbol.")
334             print("5. Análisis de datos.")
335             print("6. Guardar datos.")
336             print("7. Salir.")
337             print("")
338             opcion = input("Seleccione una opción: ")
339             print("")
340
341             if opcion == "1":
342                 mostrar_arboles()
343             elif opcion == "2":
344                 agregar_arbol()
345             elif opcion == "3":
346                 actualizar_arbol()
347             elif opcion == "4":
348                 eliminar_arbol()
349             elif opcion == "5":
350                 analisis_datos()
351             elif opcion == "6":
352                 guardar_datos()
353             elif opcion == "7":
354                 print("Salir.")
355                 break
356             else:
357                 print("Opción no válida")
358
359         print("")
360
361     menu()
362
363
364
365
366
367
368
369
370
371

```

8. CONCLUSIONES

El proyecto "Localización de Especies de Árboles Endémicas en el Municipio de Tuta" destaca la importancia de preservar la biodiversidad local mediante la creación de una plataforma web accesible y educativa.

A través de esta herramienta, la comunidad de Tuta podrá reconocer y valorar las especies de árboles endémicos, comprendiendo su papel crucial en el equilibrio de los ecosistemas y su resistencia a las variaciones climáticas.