

Informe Cocreación 3

Contenido

Index:	2
Cundinamarca y Boyacá	3
Biodiversidad Cundinamarca:	5
Biodiversidad Boyacá:	7
Datos biodiversidad:	12

Index:

1. Doctype y Estructura Básica:

<!DOCTYPE html>: Declara el tipo de documento y la versión de HTML.

<html lang="en">: El atributo lang especifica el idioma del contenido

2. Metadatos y Enlaces:

<meta charset="UTF-8">: Establece la codificación de caracteres en UTF-8.

<meta name="viewport" content="width=device-width, initial-scale=1.0">: Asegura que la página sea responsive y se ajuste a diferentes tamaños de pantalla.

<title>Biodiversidad</title>: Define el título de la página que aparece en la pestaña del navegador.

3. Estilos CSS:

Se utilizan @import para incluir archivos CSS específicos basados en el tamaño de la pantalla:

- style.css para pantallas grandes (min-width: 1024px).
- stylemovil.css para pantallas pequeñas (max-width: 600px).
- stylemedio.css para pantallas medianas (min-width: 601px y max-width: 1023px).

4. Contenido de la Página:

<header>: Contiene el título principal de la página.

<input type="checkbox" id="menu"> y <label for="menu" id="labelmenu">: Utilizados para crear un menú desplegable, generalmente para el diseño móvil.

<nav>: Contiene un menú de navegación con enlaces a diferentes páginas y recursos externos.

<section id="introduccion">: Sección principal con información sobre biodiversidad, incluyendo un artículo con texto, imágenes y enlaces relevantes.

<footer>: Incluye información de derechos reservados, enlaces a sitios relacionados, íconos de redes sociales y un logotipo.

5. Enlaces y Recursos:

- Enlaces a páginas de información adicional sobre biodiversidad y a recursos de instituciones relevantes.
- Íconos y logotipo referenciados mediante rutas a archivos locales en assets/img.

6. Elementos Multimedia:

: Utilizado para incluir imágenes dentro del contenido y del pie de página.

Cundinamarca y Boyacá

1. Elementos de Estructura y Diseño

<header>: Contiene el título principal de la página (<h1>), que en este caso es "Biodiversidad de Cundinamarca".

<input type="checkbox" id="menu"> y <label for="menu" id="labelmenu">: Estos elementos implementan un menú de navegación móvil accesible mediante un checkbox. El label actúa como el botón del menú.

<nav>: Contiene la navegación principal del sitio con una lista de enlaces a diferentes secciones y páginas externas.

2. Secciones de Contenido

<section id="introduccion">: Proporciona una introducción a la biodiversidad de Cundinamarca. Incluye un párrafo descriptivo y una imagen que ilustra la región.

<section> (con id "mapainteractivo"): Contiene un mapa interactivo de Cundinamarca. Incluye un mapa con áreas definidas que activan funciones de JavaScript cuando se pasa el ratón sobre ellas (onmouseover).

<section id="flora">: Detalla la flora de Cundinamarca. Incluye una galería de imágenes relacionadas con la flora y párrafos descriptivos sobre las plantas.

<section id="fauna">: Describe la fauna de Cundinamarca, incluyendo mamíferos, aves, reptiles y anfibios. Incluye un video de YouTube sobre fauna.

<section id="ecosistema">: Explica los diversos ecosistemas en Cundinamarca, como bosques andinos y páramos. Contiene una galería de imágenes relacionadas con los ecosistemas.

<section id="conservacion">: Detalla las estrategias de conservación en Cundinamarca, incluyendo áreas protegidas y programas de reforestación. Incluye un video sobre conservación ecoturística.

<section id="contacto">: Proporciona un formulario de contacto para que los usuarios puedan enviar consultas. Incluye campos para nombre, correo electrónico, asunto y mensaje.

3. Elementos de Multimedia y Multimedia Interactiva

Galerías de Imágenes: Utilizadas en las secciones de flora y ecosistema para mostrar imágenes relevantes. Cada imagen está dentro de un contenedor <div> con la clase "imagen".

<iframe>: Integración de videos de YouTube en las secciones de fauna y conservación para proporcionar contenido multimedia adicional.

4. Pie de Página (<footer>)

Derechos Reservados: Un aviso de copyright y derechos reservados para la página.

Enlaces de Instituciones: Enlaces a sitios web de Universidad Sergio Arboleda y el Ministerio de las TIC.

Redes Sociales: Enlaces a íconos de redes sociales como Facebook, Instagram y YouTube.

Logo: Muestra el logo

5. Otros Elementos Técnicos

<style>: Inclusión de estilos CSS para diferentes resoluciones de pantalla mediante @import para manejar la responsividad.

<script src="assets/js/region.js">: Incluye un archivo JavaScript externo para manejar interacciones y eventos en la página.

6. Importancia de Cada Elemento

Estructura y Diseño: Establecen la apariencia y la estructura general del sitio, asegurando que la navegación sea clara y accesible.

Secciones de Contenido: Proporcionan información detallada sobre diferentes aspectos de la biodiversidad y conservación en Cundinamarca.

Multimedia y Multimedia Interactiva: Mejoran la experiencia del usuario al proporcionar imágenes y videos relacionados con el contenido.

Pie de Página: Ofrece información adicional sobre derechos y enlaces relevantes, además de proporcionar acceso a redes sociales.

Elementos Técnicos: Aseguran la correcta visualización y funcionalidad del sitio en diferentes dispositivos y navegadores.

Biodiversidad Cundinamarca:

```
#web scraping utilizando la libreria BeautifulSoup
#librerias para realizar la rutina de webscrapping
import requests
from bs4 import BeautifulSoup
#url donde se va a extraer la información de la pagina
url="https://mapas1.cundinamarca.gov.co/dependencias/secambiente/noticias/cundinamarca+pulmon+diverso+del+centro+del+pais"
#respuesta de petición a la url que es la pagina
response = requests.get(url)
#llamado a la libreria BeautifulSoup
soup = BeautifulSoup(response.text, 'html.parser')
#Extraer todos las etiquetas p de la pagina utilizando el selector
elementos = soup.select("p")
#lista vacia donde se guardara la informacion extraida de la pagina
info_biodiversidad_cundinamarca = []
#####Analizar y filtrar la informacion#####
for i in range(len(elementos)):
    elemento = elementos[i]
    texto = elemento.text.replace(" ", "").strip()
    if not(texto.strip() == "") and not(i>25):
        info_biodiversidad_cundinamarca.append(texto)
#####Imprimir la informacion extraida#####
for info in info_biodiversidad_cundinamarca:
    print(info)
```

1. Importación de Librerías

```
import requests
from bs4 import BeautifulSoup
```

- **requests:** Se utiliza para enviar solicitudes HTTP a la página web desde la que se desea extraer datos.
- **BeautifulSoup:** Es una biblioteca que permite analizar documentos HTML y XML, facilitando la extracción de datos.

2. Definición de la URL

```
url =
"https://mapas1.cundinamarca.gov.co/dependencias/secambiente/noticias/cundina
marca+pulmon+diverso+del+centro+del+pais"
```

- Aquí se define la URL del sitio web del que queremos extraer la información.

3. Solicitud a la Página Web

```
response = requests.get(url)
```

- Se envía una solicitud GET a la URL y almacenamos la respuesta en la variable response. Esta respuesta contiene el contenido HTML de la página.

4. Parseo del HTML

```
soup = BeautifulSoup(response.text, 'html.parser')
```

- Se utiliza BeautifulSoup para analizar el contenido HTML de la página. response.text contiene el HTML en formato de texto, y 'html.parser' es el analizador que se usa para interpretar este HTML.

5. Extracción de Elementos HTML

```
elementos = soup.select("p")
```

- Utilizamos el método select para extraer todos los elementos <p> (párrafos) de la página. Esto devuelve una lista de objetos BeautifulSoup que representan estos elementos.

6. Inicialización de la Lista

```
info_biodiversidad_cundinamarca = []
```

- Creamos una lista vacía para almacenar la información extraída de los párrafos.

7. Filtrado y Limpieza de Datos

```
for i in range(len(elementos)):
    elemento = elementos[i]
    texto = elemento.text.replace("•", "").strip()
    if not(texto.strip() == "") and not(i >= 25):
        info_biodiversidad_cundinamarca.append(texto)
```

- Iteramos sobre cada elemento de la lista elementos:
 - **elemento.text**: Extrae el texto de cada párrafo.
 - **.replace("•", "")**: Elimina el carácter "•" del texto.
 - **.strip()**: Elimina espacios en blanco al principio y al final del texto.
 - **if not(texto.strip() == "") and not(i >= 25)**: Filtra los párrafos vacíos y solo procesas hasta el párrafo 25 para evitar sobrecargar el análisis (esto es útil si la página tiene muchos párrafos y solo te interesa un subconjunto).

8. Impresión de la Información Extraída

```
for info in info_biodiversidad_cundinamarca:  
    print(info)
```

- Iteramos sobre la lista `info_biodiversidad_cundinamarca` y se imprime cada elemento. Esto muestra el texto de los párrafos extraídos y filtrados.

Biodiversidad Boyacá:

[illegible]

1. Importación de librerías:

```
import requests
from bs4 import BeautifulSoup
```

- requests se usa para hacer solicitudes HTTP y obtener el contenido de la página web.
- BeautifulSoup se utiliza para analizar el contenido HTML y extraer la información deseada.

2. Definición de la URL y obtención de la respuesta:

```
url="https://cifras.biodiversidad.co/boyaca"
response = requests.get(url)
```

- Se define la URL de la página web de la que se desea extraer información.

- requests.get(url) realiza una solicitud HTTP GET para obtener el contenido de la página.

3. Creación del objeto BeautifulSoup:

```
soup = BeautifulSoup(response.text, 'html.parser')
```

- Se crea un objeto BeautifulSoup para analizar el contenido HTML de la respuesta.

4. Inicialización de una lista para almacenar la información:

```
info_biodiversidad = []
print("Biodiversidad de Boyaca")
```

5. Extracción de datos específicos:

- **Número de especies estimadas:**
 - selector_especies_estimadas = "#__next > main > div.bg-cover.bg-center.pt-8.lg\:pt-14.pb-3\.5.h-\[550px\] > div > div.flex.flex-col.md\:flex-row.max-h-48.justify-between.gap-y-4.w-10\12.mx-auto.-mt-9.md\:-mt-0 > div.w-1\2.relative.flex.items-center > div.left-\[15rem\].absolute.w-full.md\:w-1\3.lg\:w-full.text-white.flex.-space-y-1.flex-col.-top-\[9\%\].lg\:-top-\[8\%\].left-\[15rem\].md\:left-60 > span"


```
especies_estimadas =
soup.select_one(selector_especies_estimadas)
info_biodiversidad.append("Especies estimadas: "+
especies_estimadas.text)
```
 - selector_especies_estimadas es un selector CSS que identifica el elemento HTML con el número estimado de especies.
 - soup.select_one(selector_especies_estimadas) selecciona ese elemento.
 - La información extraída se agrega a la lista info_biodiversidad.
- **Número de especies observadas:**
 - selector_especies_observadas = "#__next > main > div.bg-cover.bg-center.pt-8.lg\:pt-14.pb-3\.5.h-\[550px\] > div > div.flex.flex-col.md\:flex-row.max-h-48.justify-between.gap-y-4.w-10\12.mx-auto.-mt-9.md\:-mt-0 > div.w-1\2.relative.flex.items-center > div.left-\[15rem\].absolute.text-yellow-green.flex.flex-col.-space-y-1.w-full.md\:w-1\3.lg\:w-full.top-\[38\%\].md\:top-\[40\%\].lg\:top-\[35\%\].md\:left-60 > span.text-lg.lg\:text-4xl.font-black.font-inter"


```
especies_observadas =
soup.select_one(selector_especies_observadas)
```



```
info_biodiversidad.append("Especies observadas: "+
especies_observadas.text)
```

- Similar al anterior, pero para el número de especies observadas.

- **Número de especies publicadas:**

- ```
selector_especies_publicadas="#__next > main > div.bg-cover.bg-
center.pt-8.lg\:pt-14.pb-3\.5.h-\[550px\] > div > div.flex.flex-
col.md\:flex-row.max-h-48.justify-between.gap-y-4.w-10V12.mx-auto.-
mt-9.md\:mt-0 > div.md\:w-1V2.my-auto.h-40.border-t-2.md\:border-l-
2.md\:border-t-0.border-yellow-green.border-dotted.flex.items-
center.py-2\.5 > div > p"
especies_publicadas =
soup.select_one(selector_especies_publicadas)
info_biodiversidad.append("Especies publicadas: "+
especies_publicadas.text)
```

- Similar a los anteriores, pero para el número de especies publicadas.

- **Información general:**

- ```
select_infos="#__next > main > div.py-8.grid.grid-cols-2.md\:grid-
cols-3.lg\:grid-cols-5.max-w-screen-xl.mx-auto.w-10V12 p"
info_general = soup.select(select_infos)
for info in info_general:
    if not (info.text in "gallery tooltip"):
        info_biodiversidad.append(info.text)
```
- Se usa un selector CSS para obtener varios elementos p (párrafos) de la página.
- Se recorren esos elementos y se filtra el texto para evitar ciertos valores.

6. Impresión de la información:

```
for info in info_biodiversidad:
    print(info)
```

- Finalmente, se imprime cada elemento de la lista info_biodiversidad.


```
selector_especies_estimadas = "#__next > main > div.bg-cover.bg-center.pt-8.lg\:pt-14.pb-3\5.h-\[550px\] > div > div.flex.flex-col.md\:flex-row.max-h-48.justify-between.gap-y-4.w-10\12.mx-auto.-mt-9.md\:.-mt-0 > div.w-1\2.relative.flex.items-center > div.left-\[15rem\].absolute.w-full.md\:w-1\3.lg\:w-full.text-white.flex.-space-y-1.flex-col.-top-\[9%\].lg\:.-top-\[8%\].left-\[15rem\].md\:left-60 > span"
especies_estimadas = soup.select_one(selector_especies_estimadas)
print(f"Especies estimadas: {especies_estimadas.text}")
```

- selector_especies_estimadas es un selector CSS que identifica el elemento con el número estimado de especies.
- soup.select_one(selector_especies_estimadas) selecciona ese elemento.
- print(f"Especies estimadas: {especies_estimadas.text}") imprime el número de especies estimadas.

Número de especies observadas:

```
selector_especies_observadas = "#__next > main > div.bg-cover.bg-center.pt-8.lg\:pt-14.pb-3\5.h-\[550px\] > div > div.flex.flex-col.md\:flex-row.max-h-48.justify-between.gap-y-4.w-10\12.mx-auto.-mt-9.md\:.-mt-0 > div.w-1\2.relative.flex.items-center > div.left-\[15rem\].absolute.text-yellow-green.flex.flex-col.-space-y-1.w-full.md\:w-1\3.lg\:w-full.top-\[38%\].md\:top-\[40%\].lg\:top-\[35%\].md\:left-60 > span.text-lg.lg\:text-4xl.font-black.font-inter"
especies_observadas = soup.select_one(selector_especies_observadas)
print(f"Especies observadas: {especies_observadas.text}")
```

- selector_especies_observadas es un selector CSS para el número de especies observadas.
- Similar al anterior, se selecciona e imprime este valor.

Número de especies publicadas:

```
selector_especies_publicadas = "#__next > main > div.bg-cover.bg-center.pt-8.lg\:pt-14.pb-3\5.h-\[550px\] > div > div.flex.flex-col.md\:flex-row.max-h-48.justify-between.gap-y-4.w-10\12.mx-auto.-mt-9.md\:.-mt-0 > div.md\:w-1\2.my-auto.h-40.border-t-2.md\:border-l-2.md\:border-t-0.border-yellow-green.border-dotted.flex.items-center.py-2\5 > div > p"
especies_publicadas = soup.select_one(selector_especies_publicadas)
print(f"Especies publicadas: {especies_publicadas.text}")
```

- selector_especies_publicadas es un selector CSS para el número de especies publicadas.
- Se selecciona e imprime este valor.

Información general:

```
select_infos="#__next > main > div.py-8.grid.grid-cols-2.md\:grid-cols-3.lg\:grid-
cols-5.max-w-screen-xl.mx-auto.w-10\12 p"
info_general = soup.select(select_infos)
for info in info_general:
    if not (info.text in "gallery tooltip"):
        print("-",info.text)
```

- select_infos es un selector CSS para obtener varios párrafos (<p>) de la página.
- Se recorren esos párrafos y se filtra el texto para evitar valores no deseados como "gallery tooltip".
- Cada texto filtrado se imprime precedido por un guion.

Datos biodiversidad:

1. Importación de librerías:

```
import pandas as pd # Librería para manejar datos en tablas (DataFrames)
import matplotlib.pyplot as plt # Librería para hacer gráficos
```

2. Datos sobre especies:

Se definen dos conjuntos de datos:

Especies endémicas: Especies que solo se encuentran en ciertos municipios.

Especies amenazadas: Especies que están en peligro en ciertos municipios.

```
data_especies_endemicas = {
    'municipio': ['La Calera', 'Subachoque', 'Guasca', 'Fomeque', 'Soacha',
                  'Fusagasuga', 'San Antonio del Tequendama', 'Choachi', 'San Francisco',
                  'Facatativa'],
    'num_especies': [197, 178, 167, 152, 134, 120, 119, 114, 112, 99],
}
data_especies_amenazadas = {
    'municipio': ['Guasca', 'La Calera', 'Soacha', 'Fusagasuga', 'San Antonio
                  del Tequendama', 'Subachoque', 'Fomeque', 'Silvania', 'Medina', 'San
                  Francisco'],
    'num_especies': [44, 43, 41, 40, 39, 38, 33, 32, 30, 28],
}
```

3. Creación de DataFrames:

DataFrame es una estructura de datos similar a una tabla en pandas.

```
df_endemicas = pd.DataFrame(data_especies_endemicas)
df_amenazadas = pd.DataFrame(data_especies_amenazadas)
```

4. Cálculo de estadísticas:

Se suman y promedian los números de especies para cada tipo (endémicas y amenazadas).

```
total_endemicas = df_endemicas['num_especies'].sum()
total_amenazadas = df_amenazadas['num_especies'].sum()
prom_endemicas = int(df_endemicas['num_especies'].mean())
prom_amenazadas = int(df_amenazadas['num_especies'].mean())
num_endemicas = df_endemicas['num_especies'].count()
num_amenazadas = df_amenazadas['num_especies'].count()
```

5. Impresión de resultados:

Se muestran los datos y las estadísticas calculadas.

```
print(f"*****Especies endemicas {num_endemicas} *****")
print(df_endemicas)
print(f"Total especies endemicas {total_endemicas}")
print(f"Promedio especies endemicas {prom_endemicas}")
print()
print("*****")
print(f"*****Especies amenazadas {num_amenazadas} *****")
print(df_amenazadas)
print(f"Total especies amenazadas {total_amenazadas}")
print(f"Promedio especies amenazadas {prom_amenazadas}")
```

6. Generación de gráfico:

Se crea un gráfico de barras que muestra el número de especies endémicas en cada municipio.

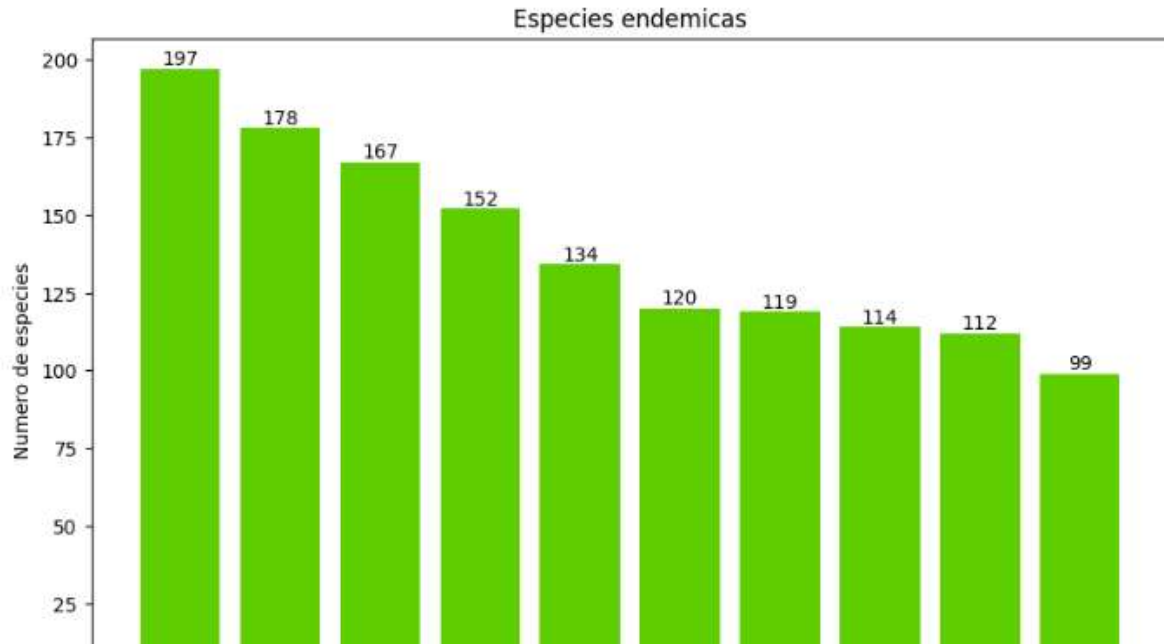
```
plt.figure(figsize=(10, 6)) # Tamaño del gráfico
colors = "#5ECE00" # Color de las barras
plt.bar(df_endemicas['municipio'], df_endemicas['num_especies'], color=colors) #
Crear el gráfico de barras
plt.xlabel('Municipio') # Etiqueta del eje X
plt.ylabel('Número de especies') # Etiqueta del eje Y
plt.title('Especies endémicas') # Título del gráfico
plt.xticks(rotation=90) # Rotar etiquetas del eje X para que se lean mejor
for i, v in enumerate(df_endemicas['num_especies']): # Añadir los valores encima
de cada barra
    plt.text(i, v, str(v), ha='center', va='bottom')
```

plt.show() # Mostrar el gráfico

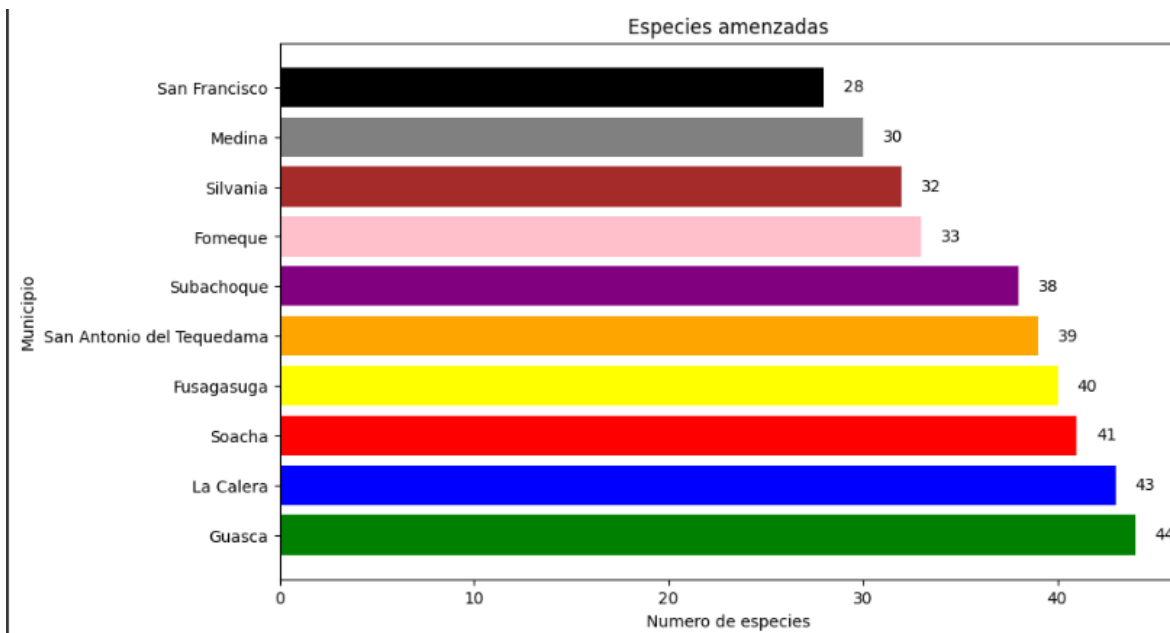
```
import pandas as pd #Libreria pandas
import matplotlib.pyplot as plt #Libreria matplotlib graficos
#data del dataframe datos biodiversidad de cundinamarca
#sitio https://cifras.biodiversidad.co/cundinamarca
#data de especies endemicas
data_especies_endemicas = {
    'municipio': ['La Calera', 'Subachoque', 'Guasca', 'Fomeque', 'Soacha',
                  'Fusagasuga', 'San Antonio del Tequendama', 'Choachi',
                  'San Francisco', 'Facatativa'],
    'num_especies': [197, 178, 167, 152, 134, 120, 119, 114, 112, 99],
}
#data de especies amenazadas
data_especies_amenazadas = {
    'municipio': ['Guasca', 'La Calera', 'Soacha', 'Fusagasuga', 'San Antonio del Tequendama',
                  'Subachoque', 'Fomeque', 'Silvania', 'Medina', 'San Francisco'],
    'num_especies': [44, 43, 41, 40, 39, 38, 33, 32, 30, 28],
}
#dataframe de especies
df_endemicas = pd.DataFrame(data_especies_endemicas)
df_amenazadas = pd.DataFrame(data_especies_amenazadas)
#calculo de datadrame
total_endemicas = df_endemicas['num_especies'].sum()
total_amenazadas = df_amenazadas['num_especies'].sum()
prom_endemicas = int(df_endemicas['num_especies'].mean())
prom_amenazadas = int(df_amenazadas['num_especies'].mean())
num_endemicas = df_endemicas['num_especies'].count()
num_amenazadas = df_amenazadas['num_especies'].count()
#imprimir resultados
print(f"*****Especies endemicas {num_endemicas} *****")
print(df_endemicas)
print(f"Total especies endemicas {total_endemicas}")
print(f"Promedio especies endemicas {prom_endemicas}")
print()
print("*****")
print(f"*****Especies amenazadas {num_amenazadas} *****")
print(df_amenazadas)
print(f"Total especies amenazadas {total_amenazadas}")
print(f"Promedio especies amenazadas {prom_amenazadas}")

#Grafica con la libreria matplotlib con barra endemica vertical
plt.figure(figsize=(10, 6)) #Ancho de 10 pulgadas y alto de 6 pulgadas
colors = "#5E9E00" #Color en hexadecimal o colores en ingles
plt.bar(df_endemicas['municipio'], df_endemicas['num_especies'], color=colors) #Bar sin h es vertical el grafico
plt.xlabel('Municipio') #Titulo del eje x
plt.ylabel('Numero de especies') #Titulo del eje y
plt.title('Especies endemicas') # Titulo del grafico
plt.xticks(rotation=90) #Rotación del titulo del eje x
for i, v in enumerate(df_endemicas['num_especies']): #Mostrar el valor en cada barra
    plt.text(i, v, str(v), ha='center', va='bottom')

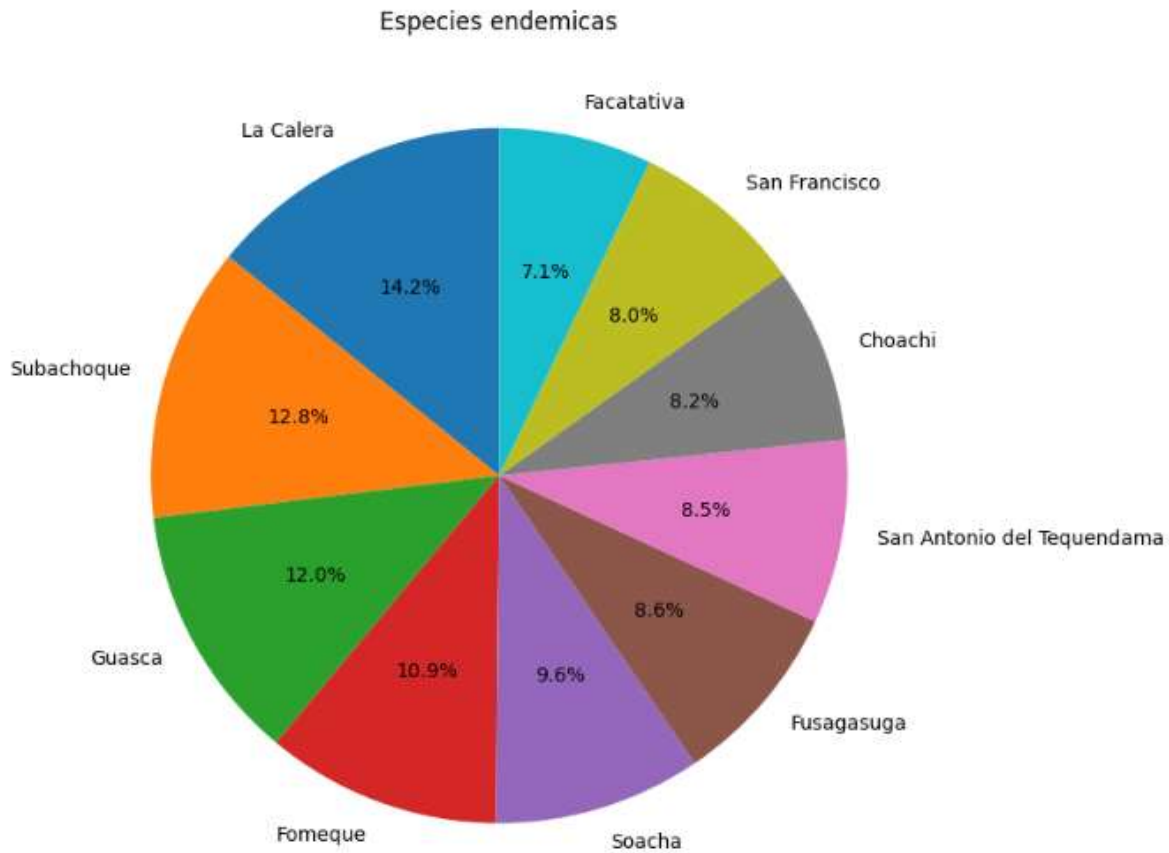
plt.show() #mostrar el grafico
```



```
#Grafica con la libreria matplotlib con barra endemica horizontal
plt.figure(figsize=(10, 6)) #Ancho de 10 pulgadas y alto de 6 pulgadas
colors = ['green', 'blue', 'red', 'yellow', 'orange', 'purple', 'pink', 'brown', 'gray', 'black']
plt.barh(df_amenazadas['municipio'], df_amenazadas['num_especies'], color=colors) #Bar sin h es vertical el grafico
plt.xlabel('Numero de especies') #Titulo del eje x
plt.ylabel('Municipio') #Titulo del eje y
plt.title('Especies amenazadas') # Titulo del grafico
for i, v in enumerate(df_amenazadas['num_especies']): #Mostrar el valor en cada barra
    plt.text(v+1, i, str(v), ha='left', va='center')
plt.show() #mostrar el grafico
```



```
#Crear grafica de pei de especies endemicas
plt.figure(figsize=(8,8))
plt.pie(df_endemicas['num_especies'], labels=df_endemicas['municipio'], autopct='%1.1f%%', startangle=90)
plt.title('Especies endemicas')
plt.show()
```



```
#Grafico de linea especies endemicas
plt.figure(figsize=(10,6))
color = 'green'
plt.plot(df_endemicas['municipio'], df_endemicas['num_especies'], marker='o', linestyle='-', color=color)
plt.xlabel('Municipio')
plt.ylabel('Numero de especies')
plt.title('Especies endemicas')
plt.xticks(rotation=90)
plt.grid(True) #Poner cuadrícula al grafico
plt.show()
```