

Informe Técnico

Proyecto "Biodiversidad CundiBoyacense"



Presentado por:

EINAR EMBUS GALINDO

Bootcamp de Programación nivel Explorador G29

Talentotech U. Sergio Arboleda

Tabla de contenido

- 1. Introducción 3
- 2. Arquitectura de la Aplicación..... 3
 - 2.1 Patrón de Diseño: MVC (Model-View-Controller) 3
 - 2.2 Estructura de Archivos..... 3
 - 2.3 Flujo de Trabajo 4
 - 2.4 Componentes Clave..... 4
- 3. Herramientas Utilizadas..... 4
 - 3.1 Lenguajes y Frameworks 4
 - 3.2 Bibliotecas y Frameworks Adicionales 5
 - 3.3 Recursos Estáticos..... 5
- 4. Desafíos Enfrentados..... 5
 - 4.1 Integración de Mapas Interactivos 5
 - 4.2 Responsividad y Compatibilidad entre Navegadores..... 6
 - 4.3 Manejo de Datos Meteorológicos 6
 - 4.4 Gestión de Contenido Dinámico..... 6
- 5. Conclusión 6

1. Introducción

Este informe técnico describe la arquitectura de la aplicación web **BioCundiBoyacense**, las herramientas utilizadas para su desarrollo y algunos de los desafíos enfrentados durante su implementación. El objetivo del proyecto es crear una plataforma interactiva que permita a los usuarios conocer y explorar la biodiversidad de las regiones de Cundinamarca y Boyacá, destacando su riqueza ecológica, cultural y natural.

2. Arquitectura de la Aplicación

2.1 Patrón de Diseño: MVC (Model-View-Controller)

La aplicación sigue el patrón de arquitectura **MVC**, que separa la lógica de la aplicación en tres componentes principales:

- **Modelos (Model):** Responsable de manejar los datos y la lógica de negocio.
- **Vistas (View):** Encargadas de la presentación y la interfaz de usuario, como se puede observar en el archivo `index.html`.
- **Controladores (Controller):** Gestiona la interacción entre el usuario, la vista y el modelo. En este caso, el archivo `app.py` actúa como controlador.

2.2 Estructura de Archivos

- **app.py:** Actúa como el controlador principal de la aplicación, donde se definen las rutas principales que gestionan las diferentes vistas y la lógica del servidor.
- **index.html:** Es una plantilla HTML que define la estructura visual de la página principal del sitio web. Utiliza **Jinja2**, un motor de plantillas que permite incluir contenido dinámico en la página.
- **Archivos estáticos:** Como las hojas de estilo CSS y las imágenes, se cargan utilizando la función `url_for('static', ...)` en el archivo HTML.

2.3 Flujo de Trabajo

1. El usuario solicita una página web
2. **Flask**, el framework utilizado en este proyecto, captura esa solicitud a través de una ruta definida en `app.py`.
3. El controlador procesa la solicitud y devuelve la vista correspondiente, utilizando plantillas HTML como `index.html`.
4. La vista puede incluir contenido dinámico que se genera en el backend y se inyecta en la plantilla mediante Jinja2.

2.4 Componentes Clave

- **Rutas:** El archivo **`app.py`** define las rutas que mapean a las diferentes secciones de la aplicación, como la página de inicio, el mapa interactivo, las estadísticas y las secciones de ecosistemas y especies.
- **Plantillas HTML:** La plantilla principal (**`index.html`**) utiliza **Bootstrap** para gestionar el diseño y la responsividad, y **Leaflet.js** para el mapa interactivo.

3. Herramientas Utilizadas

3.1 Lenguajes y Frameworks

- **Python (Flask):** Flask es el framework web ligero utilizado para gestionar el backend y las rutas de la aplicación.
- **HTML5/CSS3/JavaScript:** Son las tecnologías base utilizadas para el frontend, estructurando y estilizando las páginas del sitio web.
- **Jinja2:** El motor de plantillas que permite generar contenido HTML dinámico en base a los datos manejados en el servidor.

3.2 Bibliotecas y Frameworks Adicionales

- **Bootstrap 4.5.2:** Utilizado para crear un diseño responsivo, facilitando el manejo del layout y la estructura visual.
- **Leaflet.js:** Biblioteca JavaScript para mapas interactivos, utilizada para mostrar los puntos geográficos de interés dentro de las regiones de Cundinamarca y Boyacá.
- **OpenWeatherMap API:** Para obtener y mostrar datos meteorológicos en tiempo real en el sitio web.

3.3 Recursos Estáticos

- **Imágenes y estilos personalizados:** Se almacenan en una carpeta static, accesible desde las plantillas HTML a través de la función `url_for`.

4. Desafíos Enfrentados

4.1 Integración de Mapas Interactivos

La integración de **Leaflet.js** para crear un mapa interactivo con marcadores y popups dinámicos fue uno de los aspectos más relevantes y complejos de la aplicación. Este proceso implicó:

- La correcta implementación de los datos de coordenadas para los ecosistemas y áreas de interés.
- Asegurar que los usuarios pudieran interactuar con el mapa, haciendo zoom o clic en los marcadores para obtener más información sobre los puntos seleccionados.
- Optimizar el rendimiento del mapa en diferentes dispositivos, considerando que la aplicación debe ser accesible tanto desde dispositivos móviles como de escritorio.

4.2 Responsividad y Compatibilidad entre Navegadores

Se utilizó **Bootstrap** para garantizar que el sitio web fuera completamente responsivo. Sin embargo, adaptar los elementos visuales y los componentes del mapa a diferentes tamaños de pantalla, manteniendo su funcionalidad, presentó algunos desafíos, particularmente en dispositivos móviles de baja resolución.

4.3 Manejo de Datos Meteorológicos

La integración de la API de **OpenWeatherMap** fue otro reto debido a las restricciones de las peticiones API y la necesidad de asegurar que los datos meteorológicos se actualizarán correctamente sin afectar el rendimiento de la página web.

4.4 Gestión de Contenido Dinámico

El uso de Jinja2 para insertar contenido dinámico en las plantillas HTML fue un punto clave. Manejar correctamente los enlaces, imágenes y datos dinámicos de las diferentes secciones del portal (como ecosistemas y especies) fue crucial para garantizar una experiencia de usuario fluida.

5. Conclusión

El proyecto **BioCundiBoyacense** es una plataforma web interactiva y dinámica que combina diversas tecnologías frontend y backend para ofrecer una experiencia rica y educativa sobre la biodiversidad de las regiones de Cundinamarca y Boyacá. La arquitectura modular basada en Flask, junto con la integración de tecnologías modernas como Leaflet y Bootstrap, ha permitido crear un sitio web robusto y visualmente atractivo. Los desafíos relacionados con la responsividad, la integración de mapas y la presentación de contenido dinámico se han abordado exitosamente, resultando en una plataforma que es fácil de navegar y rica en información.