ENTREGA 5: Arquitectura Web MVC y Maquetado de Interfaz de Usuario

Objetivos de la entrega

- Incorporar patrones de arquitectura MVC para el desarrollo de clientes web livianos.
- Aplicar principios de diseño centrado en el usuario (UI/UX) para la construcción de interfaces.
- Implementar un Cliente Liviano (server side rendering).

Unidades del Programa Vinculadas

- Unidad 2: Herramientas de Concepción y Comunicación del Diseño
- Unidad 4: Diseño de Interfaz de Usuario
- Unidad 6: Diseño de Arquitectura
- Unidad 7: Integración de Sistemas
- Unidad 8: Validación del Diseño
- Unidad 9: Diseño y Seguridad

Alcance

- Implementación de un Cliente Liviano desacoplado.
- Integración con SSO.

Implementación de un Cliente Liviano desacoplado

Se deberá crear un nuevo proyecto independiente para la implementación de la interfaz gráfica de usuario correspondiente a esta instancia de MetaMapa. El cliente será desacoplado de la lógica de negocio, consumiendo las APIs REST desarrolladas en entregas anteriores.
Para la generación de las interfaces de usuario se deberá utilizar un motor de plantillas (template engine). Las vistas deberán renderizar dinámicamente los datos obtenidos de las APIs.

Requerimientos detallados

Acceso

1. El sistema debe contar con una *landing page*[1] accesible públicamente que permita:
    - Presentar el propósito y objetivos de MetaMapa.
    - Mostrar ejemplos destacados de colecciones y hechos.
    - Incluir enlaces de acceso a:
        i. Visualización de hechos sin login.
        ii. Registro o inicio de sesión para contribuyentes y administradores.
        iii. Información legal y de privacidad.
2. El sistema debe permitir el acceso anónimo para visualizar hechos y colecciones.
3. El sistema debe mostrar los hechos en un mapa interactivo y permitir ver sus detalles al hacer clic en cada marcador.

[1] El término landing page significa **página de aterrizaje o de destino**. Su misión es ser el lugar donde “aterrizan” los usuarios luego de interactuar con una campaña de atracción que los lleva a la plataforma web.

4. El sistema debe permitir el registro de contribuyentes con nombre obligatorio, y otros datos opcionales.
5. El sistema debe permitir el inicio de sesión de contribuyentes y administradores.

Visualizador

6. El usuario visualizador debe poder acceder a un listado de colecciones disponibles.
7. El usuario visualizador debe poder filtrar los hechos dentro de una colección por fecha, ubicación, categoría y fuente.
8. El usuario visualizador debe poder alternar entre modo de navegación curado e irrestricto.
9. El usuario visualizador debe poder visualizar detalles de un hecho, incluyendo ubicación, contenido multimedia y fuente.
10. El usuario visualizador debe poder generar solicitudes de eliminación de hechos con una justificación mínima de 500 caracteres.

Contribuyente

11. El contribuyente (registrado o anónimo) debe poder subir nuevos hechos a través de un formulario que permita adjuntar título, descripción, categoría, ubicación, fecha, y multimedia.
12. El contribuyente registrado debe poder editar los hechos que haya creado, dentro de un plazo de 7 días.
13. El contribuyente debe poder generar solicitudes de eliminación de hechos existentes.

Administrador

14. El administrador debe poder acceder a un panel de control con resumen de actividad (hechos, fuentes, solicitudes de eliminación).
15. El administrador debe poder crear, editar y eliminar colecciones.
16. El administrador debe poder configurar fuentes (estáticas, dinámica y proxy) asociadas a una colección.
17. El administrador debe poder configurar el algoritmo de consenso para cada colección.
18. El administrador debe poder aprobar, rechazar o aceptar con modificaciones los hechos subidos por contribuyentes.
19. El administrador debe poder aprobar o rechazar solicitudes de eliminación de hechos.
20. El administrador debe poder importar hechos desde archivos CSV con más de 10.000 entradas.

Entregables

1. **Implementación** de cliente liviano desacoplado.
2. **Bonus** - **Implementación** de un **SSO** con soporte de Social-Login.

ENTREGA 6: Despliegue, Observabilidad y Seguridad

Objetivos de la entrega

- Familiarizarse con técnicas y proveedores de despliegue.
- Incorporar herramientas de monitoreo, observabilidad y seguridad.
- Familiarizarse con diferentes estrategias de integración.
- Familiarizarse con la Arquitectura de microservicios y sus componentes asociados.

Unidades del Programa Vinculadas

- Unidad 2: Herramientas de Concepción y Comunicación del Diseño
- Unidad 6: Diseño de Arquitectura
- Unidad 8: Validación del Diseño
- Unidad 9: Diseño y Seguridad

Alcance

- Sistema desplegado en la nube.
- Herramientas de observabilidad, monitoreo, seguridad y control.
- Nuevas estrategias de diseño de APIs Web.
- Nociones de Microservicios.

Requerimientos detallados

1. Se deberá desplegar el sistema en la nube para que pueda ser accedido por el público general.
2. Se deberán implementar soluciones de observabilidad que permitan recolectar, visualizar y analizar métricas, trazas y logs del sistema, con el objetivo de obtener una visión integral y en tiempo real del estado y comportamiento de los servicios.
3. Se deberá incorporar herramientas de monitoreo proactivo que supervisen la salud de los servicios y permitan configurar políticas de autorestart ante fallos.
4. Se deberá implementar un sistema de limitación de solicitudes (Rate Limiting) que regule el tráfico entrante, protegiendo los recursos del sistema frente a abusos o sobrecarga.
5. Se deberá establecer mecanismos de control de acceso que permitan definir listas de direcciones IP autorizadas y no autorizadas, bloqueando solicitudes HTTP provenientes de fuentes no confiables o maliciosas.
6. Se deberá implementar:
    a. Opción 1: el protocolo gRPC como reemplazo de HTTP para las comunicaciones entre servicios que requieren actualizaciones en tiempo real.
    b. Opción 2: una interfaz GraphQL en el Servicio Agregador, con el objetivo de dar soporte a filtros dinámicos y selección de campos específicos, proporcionando flexibilidad y eficiencia en las consultas.
7. **Bonus -** teniendo en cuenta la arquitectura actual, ajustar los servicios actuales para que se puedan incorporar a una Arquitectura de Microservicios, puedan registrarse a un Service Registry y sean consumidos a través de un API Gateway.

Entregables

1. **Diagrama de Despliegue** de la solución actual.
2. **Implementación** de **Herramientas de Observabilidad, Monitoreo, Seguridad y Control.**
3. **Implementación de un Web Service utilizando gRPC o GraphQL.**
4. **Propuesta en diagrama de despliegue** e implementación de la Arquitectura de Microservicios.