

## ENTREGA 3: Integración, Modelado en Objetos & UX/UI

### *Objetivos de la entrega*

- Diseñar e implementar, de manera incremental, las nuevas funcionalidades.
- Exponer un servicio a través de un protocolo de red.
- Incorporar flujos de trabajo asincrónicos.
- Incorporar nociones de Diseño UI/UX.
- Reconocer la diferencia entre boceto, wireframe y maqueta.

### *Alcance*

- Implementación de una fuente propia REST.
- Integración con una fuente propia.
- Agregación de fuentes y algoritmos de consenso.
- Bocetos de las interfaces de usuario del Sistema.

### *Unidades del Programa Vinculadas*

- Unidad 2: Herramientas de Concepción y Comunicación del Diseño
- Unidad 3: Diseño con Objetos
- Unidad 4: Diseño de Interfaz de Usuario
- Unidad 6: Diseño de Arquitectura
- Unidad 7: Integración de Sistemas
- Unidad 8: Validación del Diseño

### *Dominio*

#### Modos de navegación

En iteraciones anteriores, cualquier persona podía navegar las colecciones, aplicando opcionalmente filtros. Sin embargo, en esta entrega se incorpora el concepto de **modo de navegación**, que le permitirá elegir qué hechos se mostrarán:

- de forma **irrestricta**, en la cual navegarán todos los hechos, sin ningún tipo de curación;
- de forma **curada**, en la cual se navegarán sólo los hechos para los que se tiene **consenso**.

#### Consenso

Para reducir la desconfianza que podría darse por presentar hechos dudosos o falsos, al crear una colección, se podrá especificar opcionalmente un algoritmo de **consenso**. Estos son algunos de los que se implementarán:

- **múltiples menciones**: si al menos dos fuentes contienen un mismo hecho y ninguna otra fuente contiene otro de igual título pero diferentes atributos, se lo considera consensuado;

- mayoría simple: si al menos la mitad de las fuentes contienen el mismo hecho, se lo considera consensuado;
- absoluta: si todas las fuentes contienen el mismo, se lo considera consensuado.

Si no se especifica un algoritmo de consenso, todos los hechos se consideran consensuados y estarán disponibles en la navegación curada. Por último, se considera que la ejecución de estos algoritmos puede ser costosa, por lo que es importante que no se ejecuten cada vez que ingresa un hecho sino en horarios de baja carga en el sistema.

## Exposición REST

Se deberán exponer las siguientes operaciones en una *API REST*:

### API administrativa de MetaMapa:

- Operaciones CRUD<sup>1</sup> sobre las colecciones.
- Obtención de todos los hechos de una colección.
- Modificación del algoritmo de consenso.
- Agregar o quitar fuentes de hechos de una colección.
- Aprobar o denegar una solicitud de eliminación de un hecho.

### API pública para otras instancias de MetaMapa (ver entrega anterior):

- Consulta de hechos dentro de una colección.
- Generar una solicitud de eliminación a un hecho.
- Navegación filtrada sobre una colección.
- Navegación curada o irrestricta sobre una colección.

### Requerimientos de dominio:

1. Como visualizador o contribuyente, deseo poder seleccionar el modo de navegación de los hechos.
2. Como persona administradora, quiero asociar un algoritmo de consenso a una colección.
3. El Sistema debe permitir la realización de operaciones a través de los *endpoints* solicitados, mediante el desarrollo de una API REST.

## Entregables

1. **Modelo del Dominio**: diagrama de clases que contemple las funcionalidades requeridas.
2. **Justificaciones de Diseño**: Documento y Diagramas Complementarios.
3. **Implementación** de los requerimientos de esta entrega.
4. **Diagrama de arquitectura** actualizado.
5. **Bocetos** de las principales interfaces de usuario del Sistema que sustenten los casos de uso y procesos principales.

---

<sup>1</sup> <https://developer.mozilla.org/es/docs/Glossary/CRUD>