



Universidad Tecnológica Nacional
Facultad Regional Buenos Aires
Diseño de Sistemas de Información 2025
Curso: K3101
Turno: Mañana

Trabajo Práctico Anual

Primera Entrega

GRUPO N°2	
Nombre y Apellido	Legajo
Dante Ezequiel Samudio	214.105-0
Genaro Melitón Clemente	213.006-3
José Leonardo Gutiérrez Zevallos	212.988-7
José Luis Becerra	204.675-1
Lucca Polastri	214.015-9

ENTREGA/REVISIÓN	1	2	3	4	5	6
FECHA ENTREGA	22/04/2025	27/05/2025	01/07/2025	09/09/2025	14/10/2025	25/11/2025
FIRMA DOCENTE						

Requerimientos de dominio

1. Como persona administradora, deseo crear una colección.
2. Como persona administradora, deseo poder importar hechos desde un archivo CSV.
3. Como persona visualizadora, deseo navegar todos los hechos disponibles de una colección.
4. Como persona visualizadora, deseo navegar los hechos disponibles de una colección, aplicando filtros.
5. Como persona contribuyente, deseo poder solicitar la eliminación de un hecho.
6. Como persona administradora, deseo poder aceptar o rechazar la solicitud de eliminación de un hecho.

Administración de Hechos

En esta etapa del diseño, se entiende a los hechos como unidades de información relevantes que deben ser almacenadas, buscadas y eventualmente compartidas. Cada hecho cuenta con un título, una descripción, una categoría, una ubicación geográfica, fechas asociadas (fecha del acontecimiento y fecha de carga), un origen que da cuenta de quién o qué lo reportó, y contenido multimedia vinculado. Además, se incluye un atributo booleano eliminado que permite representar la eliminación lógica del hecho sin quitarlo del sistema, conservando así trazabilidad.

Para modelar la procedencia de los hechos se definió la clase *Origen*, la cual contiene información básica sobre la fuente, como el nombre, apellido y edad del contribuyente (si aplicase) y el tipo de origen, representado por un enumerado *TipoOrigen*. Este enumerado permite distinguir entre hechos ingresados manualmente, importados desde datasets o reportados por contribuyentes.

El contenido multimedia se encapsula en la clase *Multimedia*, asociada a cada hecho, con un atributo formato representado por el enumerado *Formato* que contempla texto, imagen, audio o video. Esto asegura la extensibilidad para nuevos tipos de contenido en el futuro.

Administración de Colecciones y Filtros

Se decidió modelar la clase *Coleccion* para permitir a los usuarios administradores organizar hechos de acuerdo con un propósito determinado. Cada colección incluye un título, una descripción, un conjunto de hechos y un conjunto de filtros asociados que definen su criterio de selección.

La clase *Filtro* permite seleccionar hechos según tres criterios: la categoría, un rango de fechas y una zona geográfica. Para representar el área geográfica, se diseñó la clase *Zona*, que contiene un conjunto de ubicaciones y un método que permite determinar si un hecho pertenece a dicha zona. Este enfoque proporciona un criterio espacial flexible y extensible para búsquedas geográficas.

El método *cumpleFiltro* encapsula la lógica de validación para cada hecho respecto al filtro. Este diseño favorece la cohesión y la claridad, facilitando también futuras extensiones a nuevos tipos de criterios de filtrado.

Aunque *Filtro* no posee identidad propia dentro del sistema, su comportamiento encapsulado (especialmente a través del método *cumpleFiltro*) sugiere que es más apropiado modelarlo como una clase con lógica de negocio, que como un *value object*.



Importación y Contribución de Hechos

Se reconoció la necesidad de contar con una fuente de datos externa que permita importar hechos de manera automatizada. Para ello, se introdujo la clase *ImportadorHechos*, que encapsula la lógica necesaria para la carga de hechos desde fuentes externas, utilizando el patrón Adapter a través de la interfaz *CSVReaderAdapter*. Esta decisión permite reducir el acoplamiento con los mecanismos concretos de lectura de archivos, facilitando su extensión y reutilización.

Solicitudes de Eliminación

Dado que los hechos pueden ser reportados por diferentes fuentes, se consideró necesario incluir un mecanismo formal de revisión para las solicitudes de eliminación. Se modeló la clase *Solicitud*, que contiene referencias al hecho en cuestión, un estado (pendiente, aceptada o rechazada), y datos del responsable y supervisor de la solicitud.

El método *estaFundado* evalúa si existe fundamento para proceder con la solicitud, promoviendo así una administración responsable del contenido. Esta estructura permite incorporar un flujo de revisión que podrá expandirse a otras operaciones como ediciones o validaciones de hechos en versiones futuras.

