

El sistema diseñado permite gestionar, visualizar y organizar hechos en un mapa digital. Cada hecho representa un evento que ocurrió, con una ubicación específica (representada con latitud y longitud), un título, una descripción y su respectiva fecha.

Planteamos un diseño el cual no tiene roles de usuario, como visualizador, administrador o contribuyente. Esto debido a que es el sistema el que debe permitir a los distintos tipos de usuarios interactuar con los hechos, contribuir a nuevos eventos o filtrar la información según sus necesidades, sin asociarlos a un tipo de usuario particular.

Los hechos pueden obtenerse desde diferentes fuentes de información. Actualmente, se utiliza una fuente de datos en formato CSV, aunque el sistema está preparado para incorporar nuevas fuentes más dinámicas en el futuro. De tal manera, se le permite a los usuarios buscar hechos aplicando diferentes filtros. Además, el sistema permite crear colecciones de hechos basadas en criterios definidos, agrupando dinámicamente los eventos que cumplen con un determinado criterio.

A su vez, los usuarios pueden generar solicitudes para eliminar hechos, las cuales son marcadas con un estado de revisión (PENDIENTE), el cual puede cambiar a ACEPTADA o RECHAZADA dependiendo de la decisión del usuario administrador. Estas solicitudes contienen el título del hecho en cuestión y un motivo para su eliminación.

## Decisiones de diseño

- Decidimos utilizar el patrón “**Strategy**” con criterio para poder encapsular cada tipo de criterio particular y permitir cambiar el criterio de pertenencia, ya que es configurable.
- Decidimos utilizar el patrón “**Template**” con fuente ya que siempre se va a tener que abrir el archivo, almacenar sus hechos, y cerrar el archivo. Esa sería la plantilla. Cada forma de obtener los hechos de esas fuentes van a ser las estrategias en concreto.
- Validamos que cada atributo del constructor de hecho y colección no se cargue vacío para que no hayan inconsistencias.
- En vez de hacer una clase Ubicación, pusimos latitud y longitud como atributos.
- Pusimos Categoría como enum para dejar categorías predeterminadas y poder filtrar en la Colección de Hechos.
- Utilizamos un Enum “Estado” para representar el estado de la solicitud de eliminación.
- Creamos la interfaz criterio, de la cual heredan 3 criterios de filtrado de las colecciones.