

Explicación de casos límite

Para el manejo de casos límite del módulo Carpeta implementamos lo siguiente:
Para el caso de querer agregar una subcarpeta en la raíz y que ya exista, levantamos un ValueError. También, para el caso de agregar una subcarpeta anidada, aplica el mismo comportamiento, ya que se llama al método agregar_subcarpeta() y, si en el try-except se levanta un error indicando que ya existe, ocurre lo mismo. (Este método se llama también en la clase Usuario).

Para el caso de eliminar un mensaje, si el objeto a eliminar no es del tipo Mensaje, se levanta un TypeError, y si lo es pero no se encuentra el mensaje, se lanza un ValueError dentro del except del try-except.

Para el caso de mover un mensaje, si el destino no es un objeto de tipo Carpeta, se levanta un TypeError, y si no se encuentra el mensaje a mover, se levanta un ValueError.

Casos límite

| MÉTODO | SITUACIÓN | MANEJO IMPLEMENTADO |
|--------------------|----------------------------------------------------------------------------------------------------|------------------------|
| AGREGAR_SUBCARPETA | Agregar una subcarpeta que ya existe (tanto en la raíz como en una subcarpeta anidada) Se lanza un | Se lanza un ValueError |
| ELIMINAR_MENSAJE | Si el objeto a eliminar no corresponde al tipo mensaje | Se lanza un TypeError |
| ELIMINAR_MENSAJE | Eliminar un mensaje que no se encuentra | Se lanza un ValueError |
| MOVER_CARPETA | Mover un mensaje con un destino no válido (que no sea un objeto Carpeta) | Se lanza un TypeError |
| MOVER_CARPETA | Mover un mensaje que no existe en ninguna carpeta | Se lanza un ValueError |