

Memento

Federico Alberto Alfaro Chaverri
Cristina Urbina Cespedes
Jefferson José Salas Cordero

¿Qué hace?

El objetivo del patrón memento es capturar y externalizar el estado interno de un objeto tal que dicho objeto pueda ser restaurado a ese estado posteriormente.

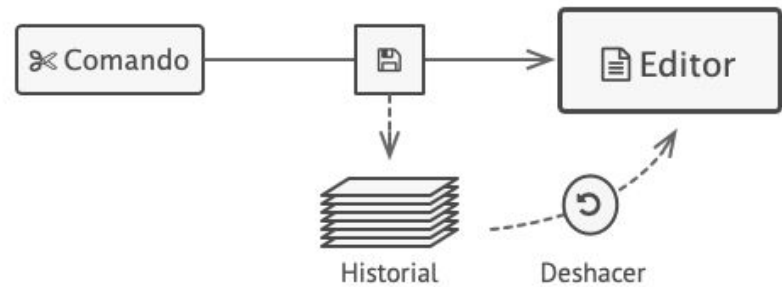


¿Cuándo aplicar el patrón memento?

- Es utilizado cuando se quiere producir instantáneo el estado del objeto para poder restaurar el estado previo del objeto.
 - Se utiliza cuando se necesitan realizar copias completas de algún objeto.
 - Es utilizado cuando se tiene un acceso directo a los campos consultores o modificadores del objeto.
-

¿Por qué se usa?

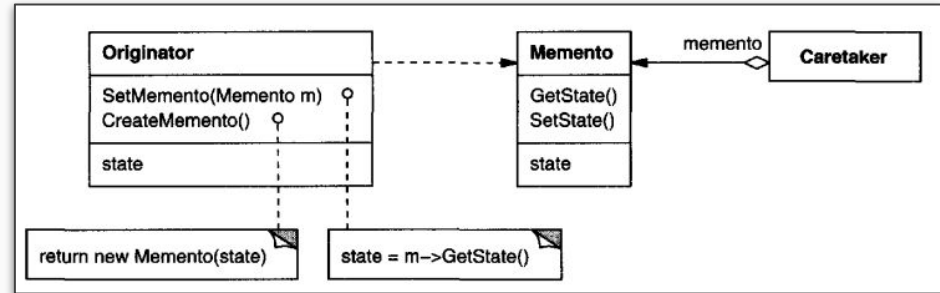
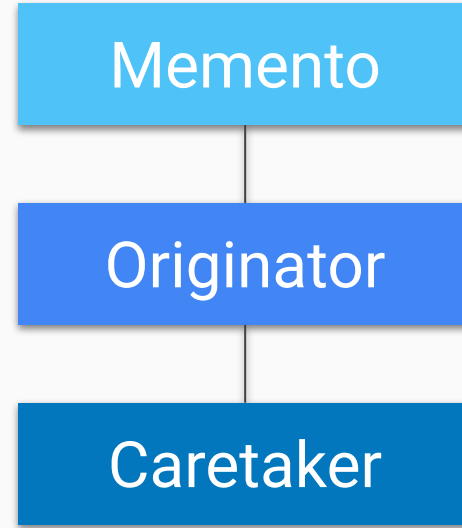
Se utiliza principalmente cuando se quiere implementar puntos de control además también se utiliza cuando se quieren hacer mecanismos que permitan a los usuarios retirarse de operaciones o recuperarse de errores



Partes que lo componen

El patrón memento se compone por 3 partes, las cuales son las siguientes:

- Memento
- Originador
- Cuidador



¿Cómo funciona?



Se debe guardar la información de un estado en algún lugar para poder restaurar los objetos a un estado previo. Esto se hace utilizando:

Memento: Almacena un snapshot del estado interno de otro objeto (el originador del memento)

Si se quiere deshacer una operación, el patrón memento funciona de la siguiente manera:

1. Cada vez que se realiza una operación cualquiera el programa le pide al originador un memento
2. El originador crea y retorna un memento el cual contiene estructuras de datos que describen el estado actual de sus ecuaciones y variables internas
3. Después cuando el usuario desea deshacer la operación, el programa devuelve el memento al originador
4. Basado en la información en el memento, el originador cambia sus estructuras internas para retornar sus ecuaciones y variables a su estado previo



Implementación

Cuando se desea implementar el patrón de memento hay que tener en consideración dos cuestiones:

- 1) Soporte de idiomas: El memento cuenta con dos interfaces, una amplia para los creadores y una estrecha para los objetos.
- 2) Almacenamiento de cambios incrementales: Cuando un memento se crea y se devuelve a su creador en una secuencia predecible. Memento puede guardar solo el cambio incremental en el estado interno del creador.

Consideraciones

- Preservación de los límites del encapsulamiento.
 - Simplificación del originador.
 - El uso de mementos puede resultar caro.
 - Definición de interfaces amplias y reducidas.
 - Costos ocultos de mantener mementos.
-

Bibliografia

Design Patterns: Elements of Reusable Object-Oriented Software. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. Pearson Education, 31 oct 1994 - 395 páginas.



Muchas gracias por su atención.