Memento

Objective

Store the internal state of an object in an environment external to it, so that this state can be restored later.

Function

Return to previous system states.

Structure

As shown in figure 1

- Client: Component that affects the Originator and records the new status with the Caretaker. In other words, it is the one who makes the change on the object and records the status.

- Originator: It is the component that changes its state.

- Memento: Component that stores the state of the Originator at a given time.

- Vigilant: Component that records the changes of the Originator. This component allows us to travel between the different states of the Originator.

beha

The structure that meets this pattern is shown in Figure 1

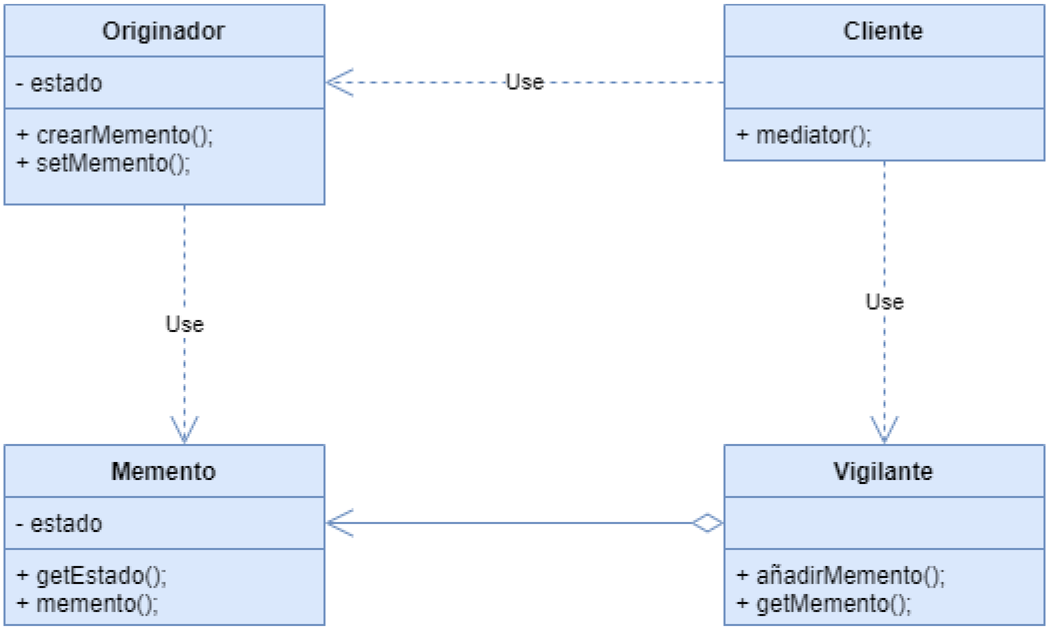

Figure 1: UML Diagram Memento Pattern

Applications

The use of the Memento pattern is recommended when:

- It is necessary to preserve the state of an object over time (persistence), and at the same time encapsulate the values of the attributes of that state of the object to external objects.

Design Patterns Collaborators

-

Scope of action

Applied at the object level.

Problem

An application wants to implement the functionality of undoing actions performed on objects, which implies saving their states; this requires the implementation of the functionality of saving and recovering the state in the object itself, saturating the memory.

Solution

The Memento design pattern suggests separating the object from the functions of saving and retrieving a particular state, while retaining the encapsulation of the object's attribute values, by restricting the access of other objects to the persistence of the object's states.

Diagram or Implementation

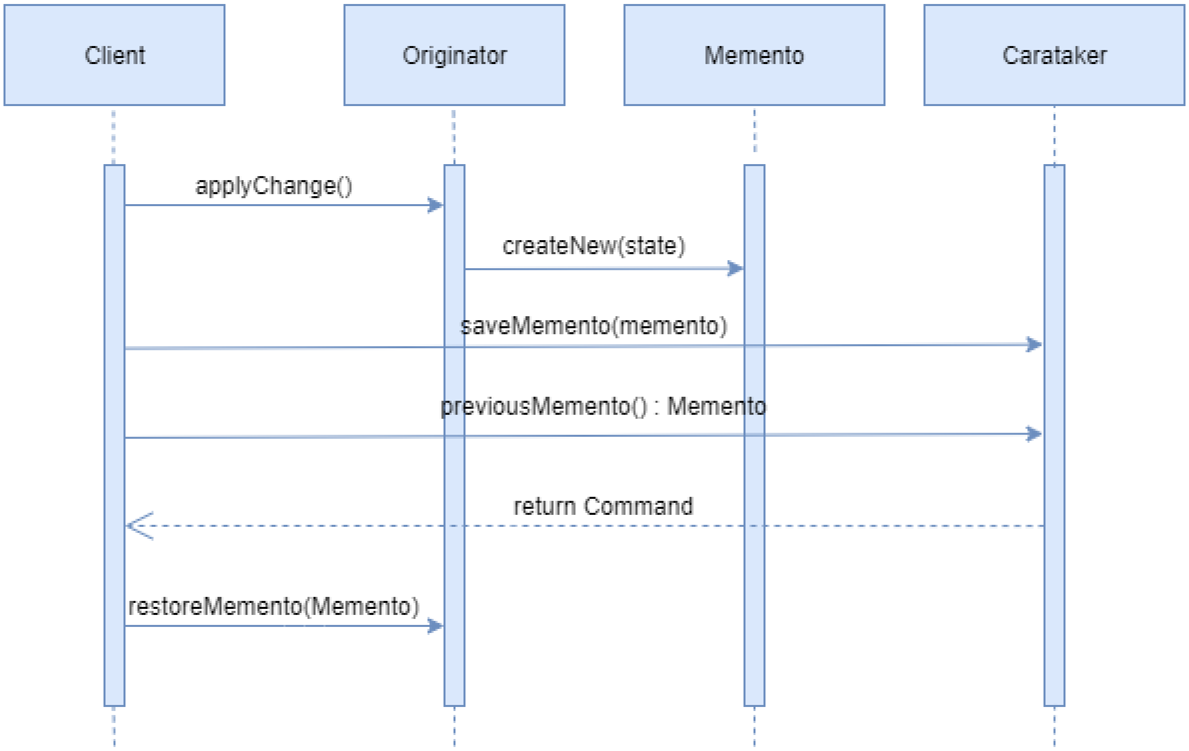

Figure 2: UML Diagram Memento Pattern

Figure 2 explains the behaviour of the pattern by means of a sequence diagram.

- The client class applies a change to the Originator component.

- The Originator component creates a new Memento that represents its current state.

- The client class saves the Memento in the Caretaker component so that it can later switch between the states of the Originator component.

- After some time, the client class asks the Caretaker component for the previous state of the Originator component.

- The client class restores the state of the Originator by means of the Memento obtained from the Caretaker component.