Patrones de Diseño de Software

PATRONES COMPORTAMIENTO - MEMENTO

De comportamiento

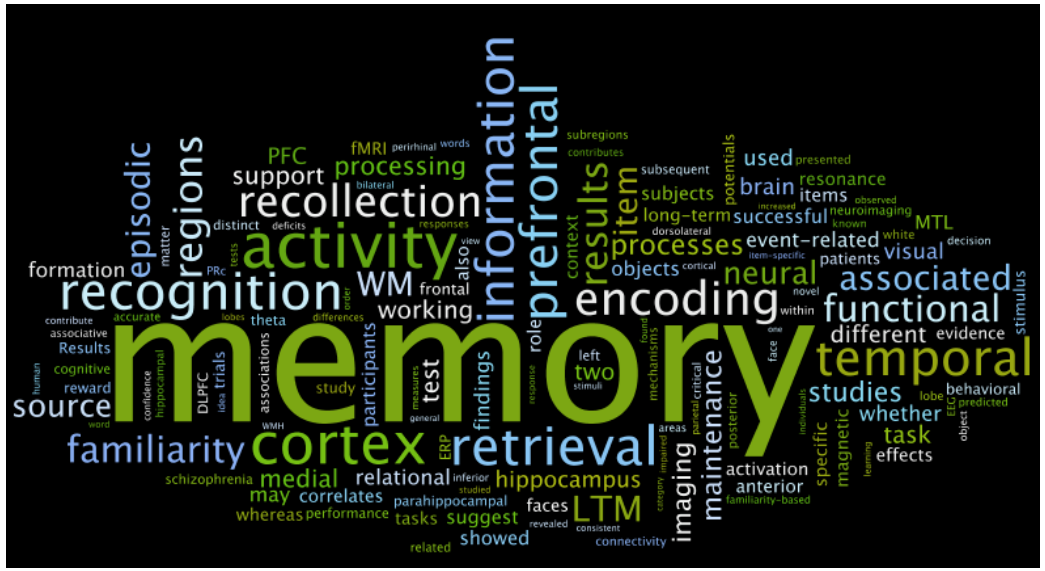
Interacción de las clases y objetos

Cadena de Responsabilidad	Una forma de pasar una petición entre una cadena de objetos
Comando	Encapsular una solicitud de comando como un objeto
Intérprete	Una forma de incluir elementos del lenguaje en un programa
Iterador	Accede secuencialmente a los elementos de una colección
Mediador	Define una comunicación simplificada entre clases
Memento	Captura y restaura el estado interno de un objeto
Observador	Una forma de notificar un cambio a una serie de clases
Estado	Altera el comportamiento de un objeto cuando su estado cambia
Estrategia	Encapsula un algoritmo dentro de una clase
Método de plantilla	Difiere los pasos exactos de un algoritmo a una subclase
Visitante	Define una nueva operación a una clase sin cambios

Patron de diseño Memento

PROPOSITO:

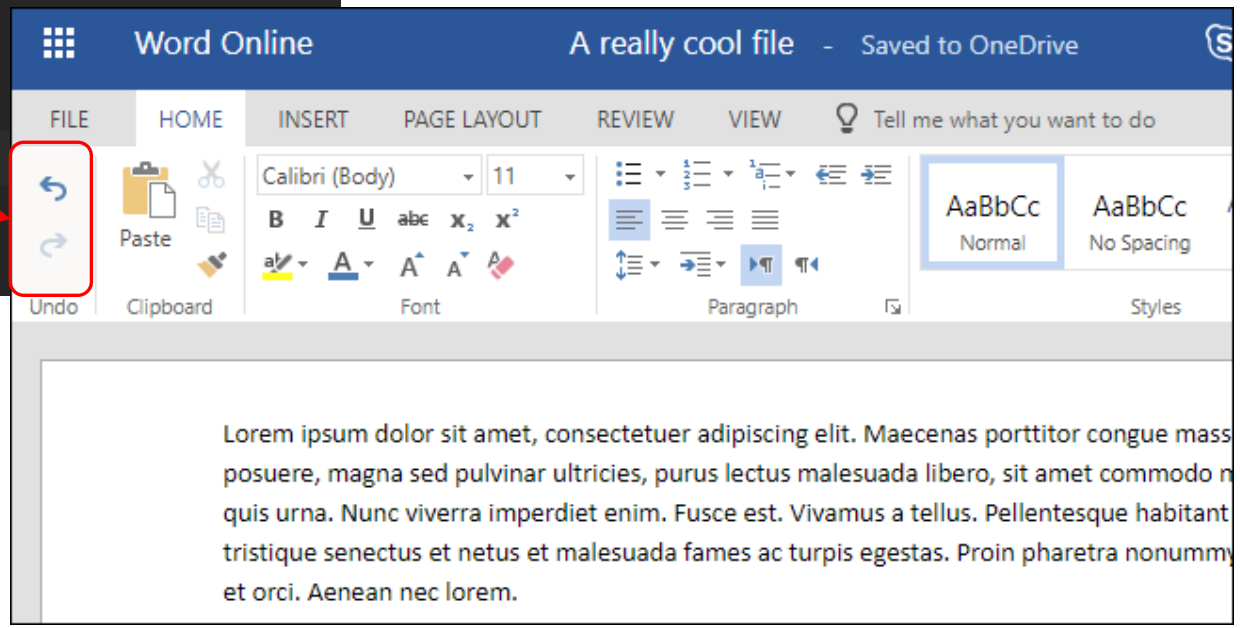
Memento es un patrón de diseño de comportamiento que te permite guardar y restaurar el estado previo de un objeto sin revelar los detalles de su implementación.



Patrones de diseño – Memento

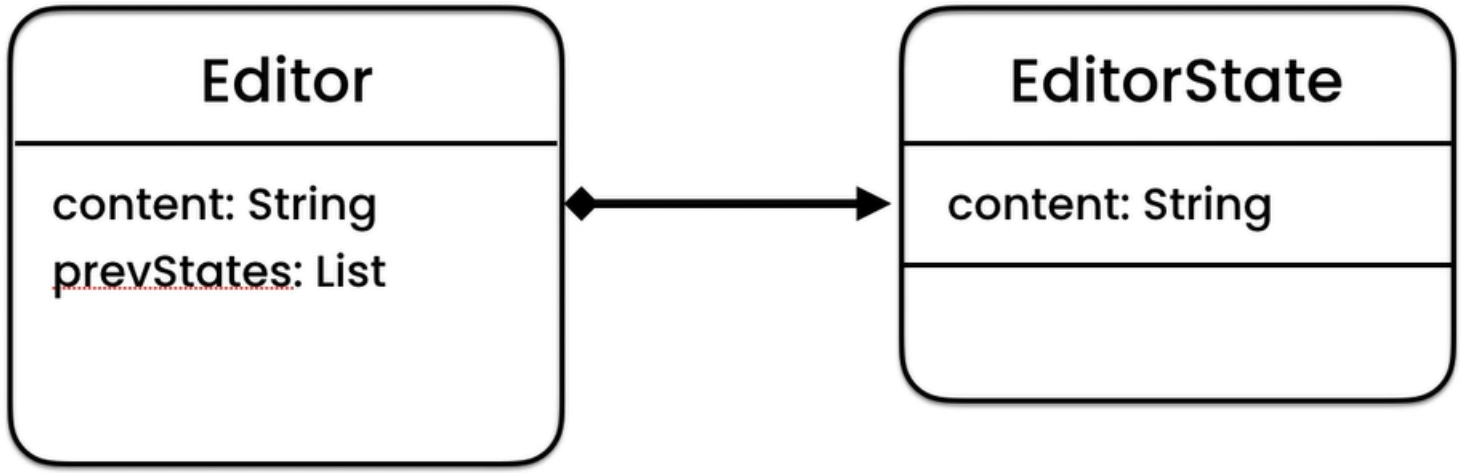
Problema: Editor de texto similar a Word

```
public class Main {
    public static void main(String[] args) {
        var editor = new Editor();
        editor.setContent("a");
        editor.setContent("b");
        editor.setContent("c");
        editor.undo();
    }
}
```



Patrones de diseño – Memento

Problema:



Patrones de diseño – Memento

Problema:

Principio de Responsabilidad Simple

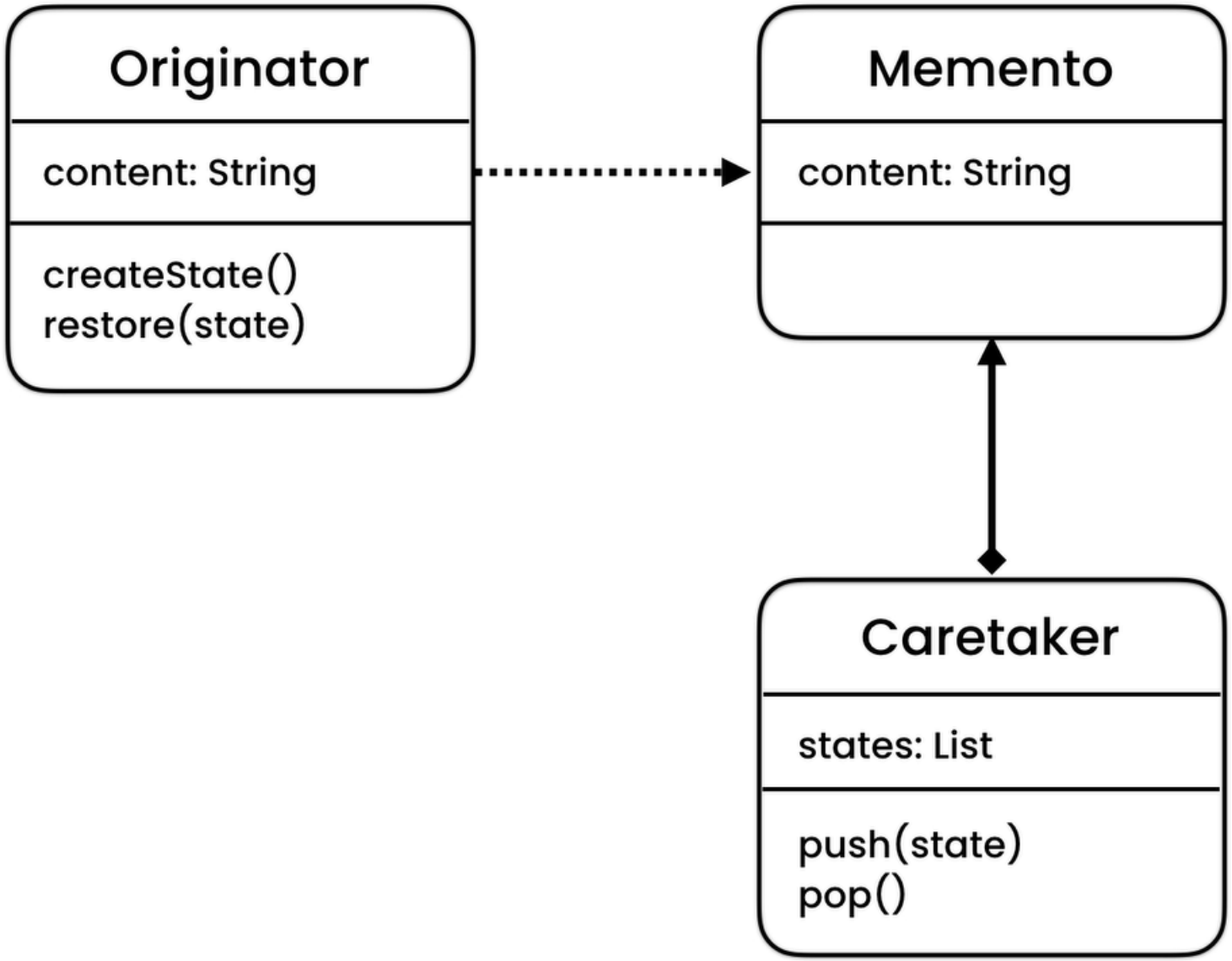
Single Responsibility Principle

Patrones de diseño – Memento

Problema:

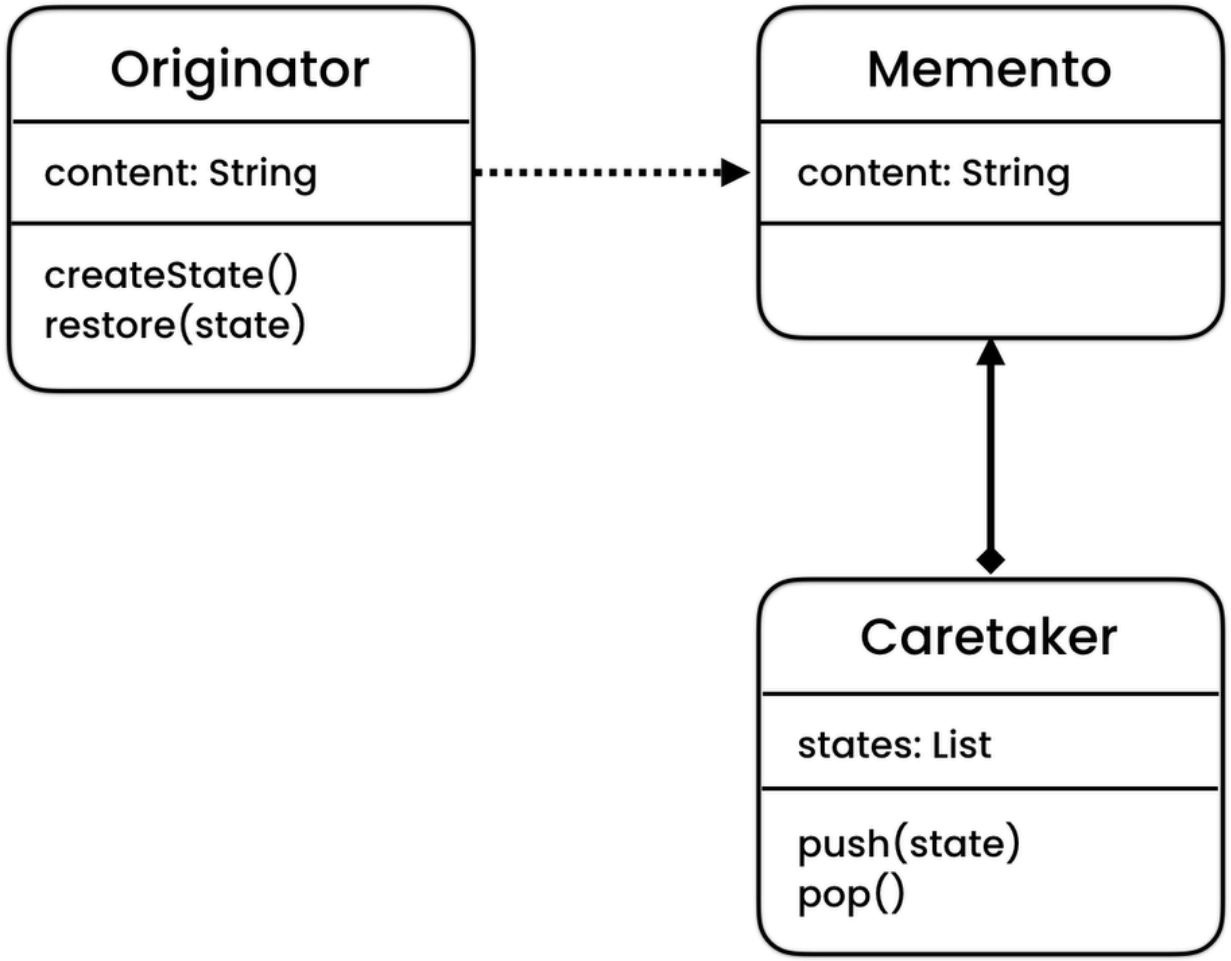
Patrones de diseño – Memento

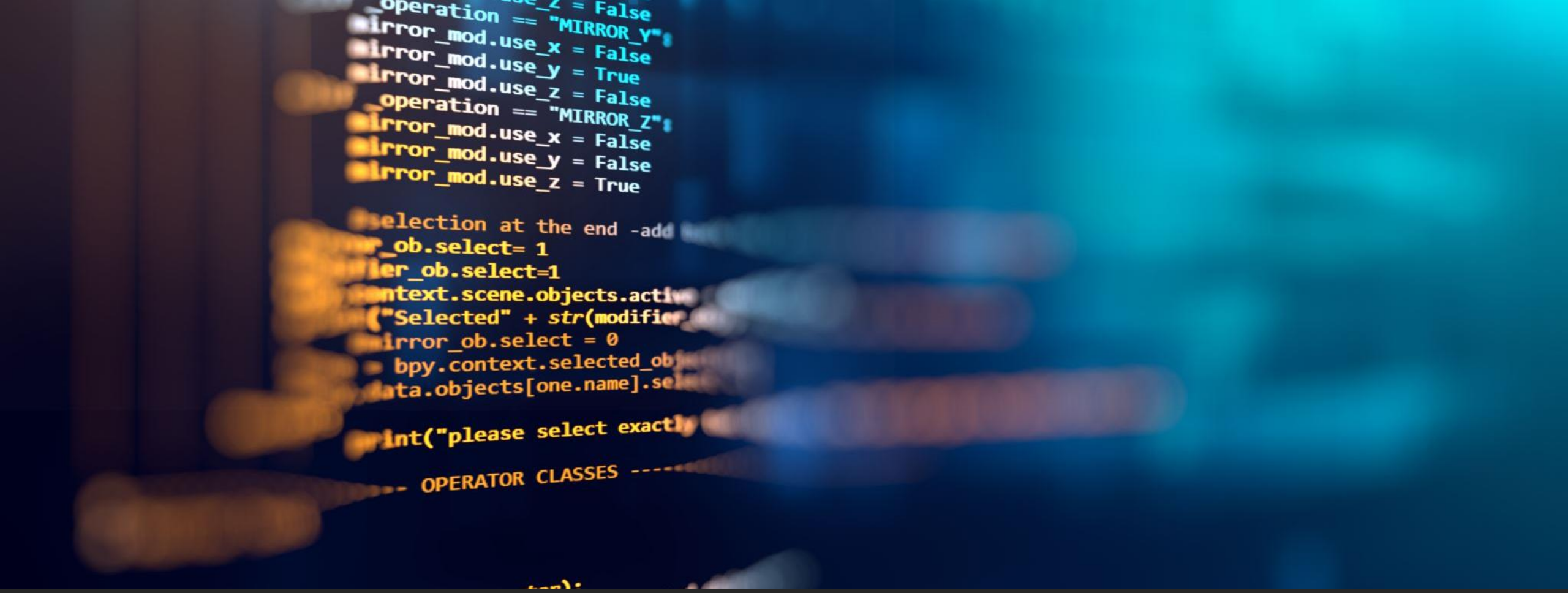
Solucion:



Patrones de diseño – Memento

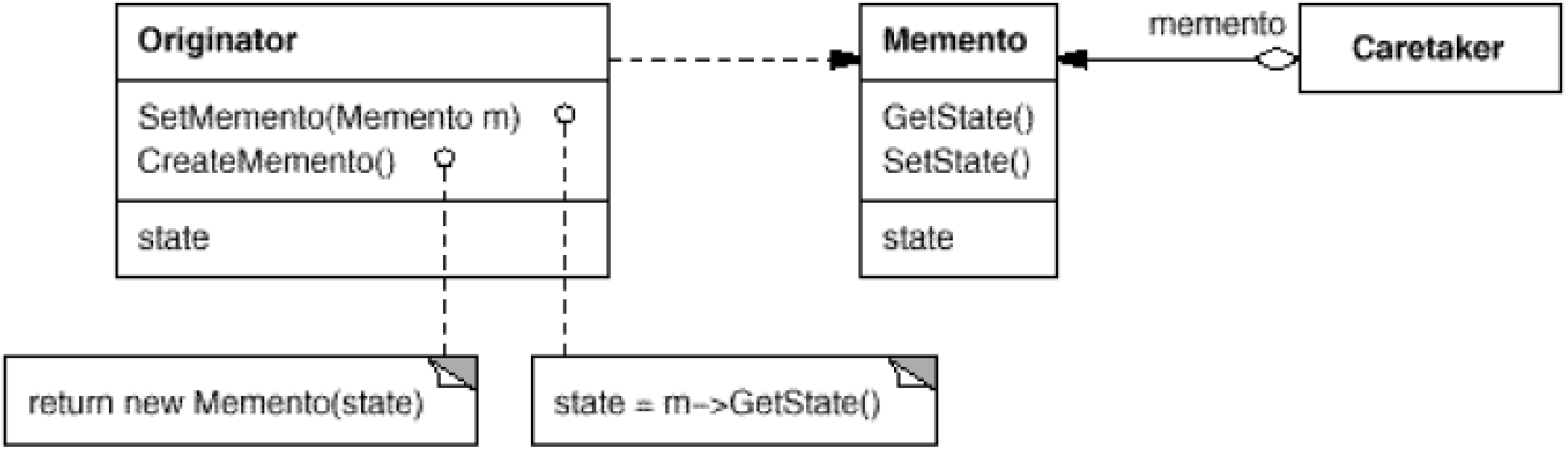
Implementación





Demo

Revisemos en Código la implementación de la solución



Patron de diseño - Memento

Estructura

Patrones de diseño – Memento

Ventajas:

- ✓ Puedes producir instantáneas del estado del objeto sin violar su encapsulación.
- ✓ Puedes simplificar el código de la originadora permitiendo que la cuidadora mantenga el historial del estado de la originadora.

Desventajas:

- × La aplicación puede consumir mucha memoria RAM si los clientes crean mementos muy a menudo.
- × Las cuidadoras deben rastrear el ciclo de vida de la originadora para poder destruir mementos obsoletos.
- × La mayoría de los lenguajes de programación dinámicos, como PHP, Python y JavaScript, no pueden garantizar que el estado dentro del memento se mantenga intacto.

Patrón de diseño Memento

Cuándo utilizar este patrón?

Utiliza el patrón Memento cuando quieras producir instantáneas del estado del objeto para poder restaurar un estado previo del objeto.

Utiliza el patrón cuando el acceso directo a los campos, consultores o modificadores del objeto viole su encapsulación.

Patrones de diseño – Memento

Práctica

Supongamos que tenemos una clase llamada Document, esta clase representa un documento en un procesador de textos como MS Word o Apple Pages.

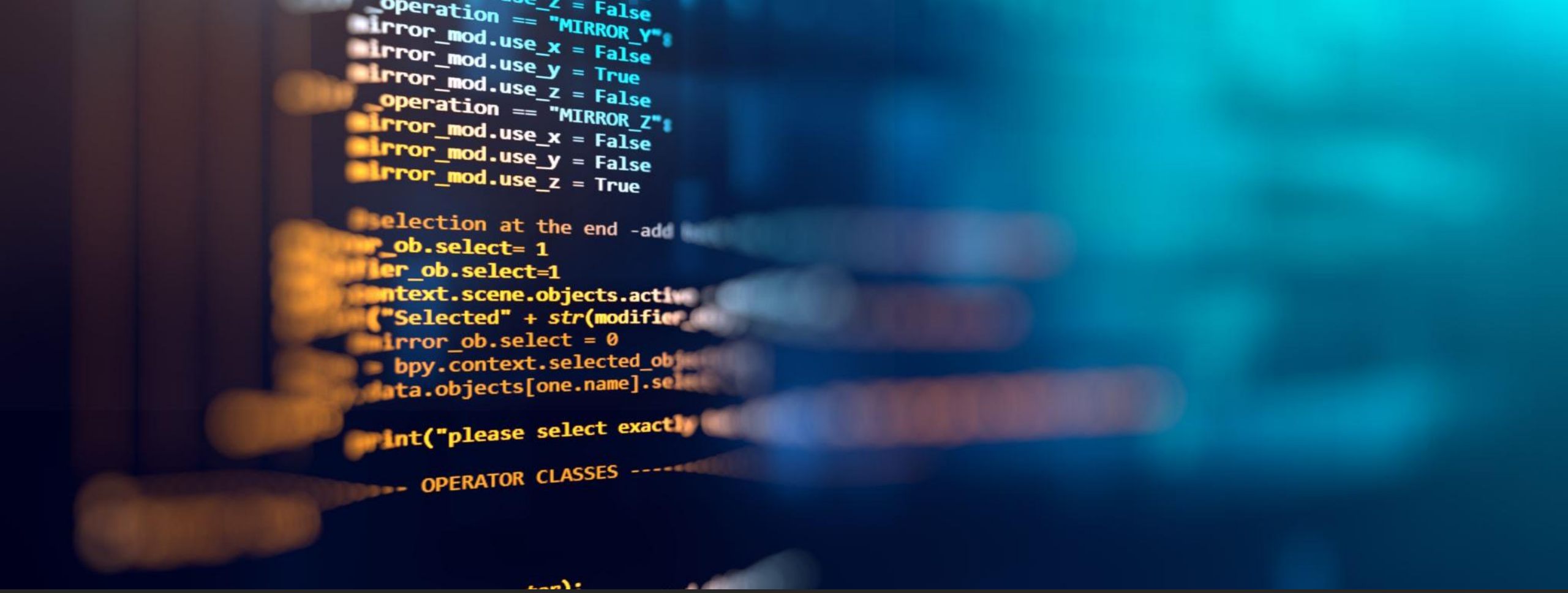
Nuestra clase Document tiene tres atributos:

- content
- fontName
- fontSize

Debemos permitir al usuario deshacer los cambios en cualquiera de estos atributos.

En el futuro, podemos añadir atributos adicionales en esta y estos atributos también deberían poder deshacerse.

Implementa la función de deshacer utilizando el patrón memento.



Demo

Solución de la tarea

Patrones de diseño – Memento

TAREA:

Supongamos que estamos desarrollando un juego como el de Mario Bros, y a medida que vamos avanzando en el juego queremos mantener el estado de nuestro juego así que vamos salvando la partida haciendo checkpoints de modo que al perder podamos recuperarnos.

Debemos almacenar el mundo, el puntaje, el id del jugador y un estado para el personaje(estrella, pluma, hongo, flor)

El jugador podrá regresar atrás las veces que desee.


Gracias