

<<Code Slayers>>

Aguirre Morales Gael Alejandro 319165704

Romero Cruz Fernando 319314256

Sanchez Nava Rodrigo 319222571

Practica 02

//State//

-Principios:

La capacidad de introducir nuevos estados sin cambiar las clases de estado existentes o el contexto.

Organizar el código relacionado con estados particulares en clases separadas

-Desventaja:

Cuando se tienen pocos estados o no cambia muy seguido puede ser excesivo aplicar el patrón

//Template//

-Principios:

Se evita la duplicación en el código usando la estructura general del flujo, se implementa una vez en el algoritmo de la clase abstracta y las variaciones necesarias se implementan en las subclases.

Flexibilidad de dejar que las subclases decidan cómo implementar los pasos

-Desventaja:

Puede llegar a ser limitado por el esqueleto de algún algoritmo

//Iterator//

-Principios:

Se puede iterar sobre la misma colección en paralelo porque cada objeto iterador contiene su propio estado de iteración.

Se pueden implementar nuevos tipos de colecciones e iteradores y pasarlos al código ya hecho sin romper algo

-Desventajas:

Aplicar el patrón puede ser exagerado dependiendo si se trabaja con colecciones simples

Usar un iterador puede ser menos eficiente, que ir a través de los elementos de alguna colección especializada directamente