Práctica de Laboratorio No. 7 Patrón Composite

Harold Andrés Velásquez Tobar
Carlos Mario Vivas García

Mgtr. Ricardo Zambrano

Universidad
del Cauca

Ingeniería de Sistemas

01/11/2023
Universidad del Cauca
Popayán – Cauca

Tabla de Contenido

Plantilla Patrón Composite

1. Desarrolle la siguiente plantilla para el patrón Composite:

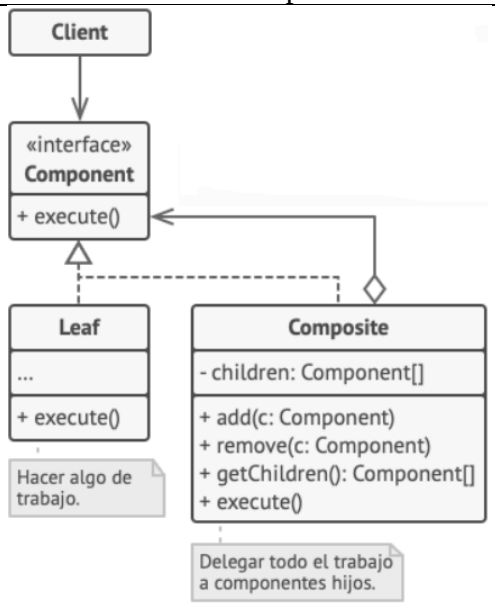
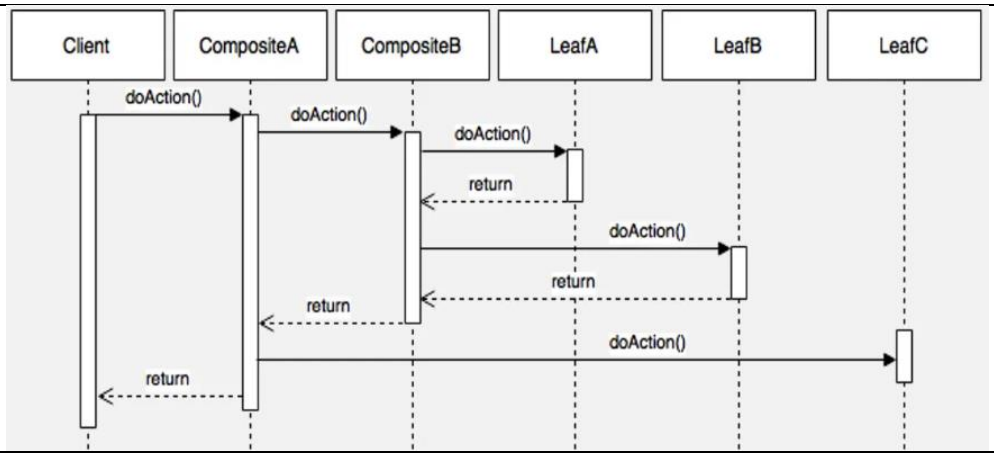
Patrón Creacional: **Composite**	
Intención	El patrón Composite se emplea en la arquitectura de software para organizar objetos en una estructura jerárquica en forma de árbol. Su principal propósito es permitir que los clientes manejen objetos individuales y composiciones de objetos de manera homogénea. La intención fundamental es crear una jerarquía que refleje de manera adecuada la relación "todo-parte" entre los objetos.
Problema que Soluciona	Cuando trabajamos con estructuras de datos jerárquicas, como árboles o grafos, es común que los clientes necesiten interactuar con todos los elementos de la jerarquía. El problema radica en que los objetos hoja (elementos individuales) y los objetos compuestos (que contienen otros objetos) pueden diferir significativamente en su estructura y comportamiento, lo que dificulta su tratamiento uniforme por parte de los clientes. El patrón Composite resuelve este problema al proporcionar una interfaz común para todos los objetos en la jerarquía, independientemente de si son hojas o compuestos.
Solución Propuesta	El patrón Composite propone la creación de una interfaz común llamada "Componente". Tanto los objetos hoja como los objetos compuestos implementan esta interfaz. Los objetos hoja representan elementos individuales de la jerarquía, mientras que los objetos compuestos pueden albergar una colección de subcomponentes que pueden ser objetos hoja o compuestos. Los objetos compuestos delegan las operaciones a sus subcomponentes, lo que posibilita una representación recursiva de la estructura jerárquica. Esto simplifica que los clientes realicen operaciones en toda la jerarquía de manera consistente a través de la interfaz del Componente.
Diagrama de Clases	

Diagrama de Secuencia	
Participantes	**Componente:** Define la interfaz común que especifica las operaciones que todos los objetos, tanto hojas como compuestos, deben implementar. **Hoja:** Representa objetos individuales que no poseen subcomponentes. Implementan la interfaz del Componente y definen sus operaciones específicas. **Compuesto:** Representa objetos que pueden contener una colección de subcomponentes, ya sean hojas u otros objetos compuestos. También implementan la interfaz del Componente y pueden gestionar la lista de subcomponentes y delegar operaciones a estos.
Aplicabilidad	El patrón Composite es útil en situaciones en las que es necesario representar estructuras jerárquicas y se busca un tratamiento uniforme de objetos individuales y sus composiciones. Algunos ejemplos de su aplicación incluyen: • Modelado de documentos con elementos anidados, como un documento HTML con etiquetas y contenido. • Representación de estructuras de árboles en interfaces gráficas de usuario, como un menú desplegable con submenús. • Implementación de estructuras de datos recursivas, como expresiones matemáticas con subexpresiones anidadas.
Consecuencias	• Facilita la incorporación de nuevos tipos de componentes sin necesidad de modificar el código del cliente, lo que mejora la extensibilidad del sistema. • Permite la construcción de estructuras complejas de manera recursiva, ya que los objetos compuestos pueden incluir otros objetos compuestos. • Simplifica el código del cliente al eliminar estructuras condicionales complejas para tratar objetos individuales y compuestos. • Puede generar sobrecarga en ciertas operaciones, dado que todas las operaciones se definen en la interfaz del Componente, incluso si algunas no son relevantes para todos los subtipos.

Implementación del Patrón Composite

Desarrolle una implementación del patrón composite en Java, que permita visualizar la estructuración de viajes continentales, de tal forma que un cliente de la agencia de viajes pueda configurar y visualizar su viaje. No es necesario crear una interfaz gráfica, pero puede ayudar.

Solución:

Enlace Diagrama UML:

https://drive.google.com/file/d/1YzAtryYSKIt9Wzj-4QO7W7UsyHejhw6p/view?usp=sharing

Funcionamiento del Código:

Figura 1. Paquetes por defecto Creados.

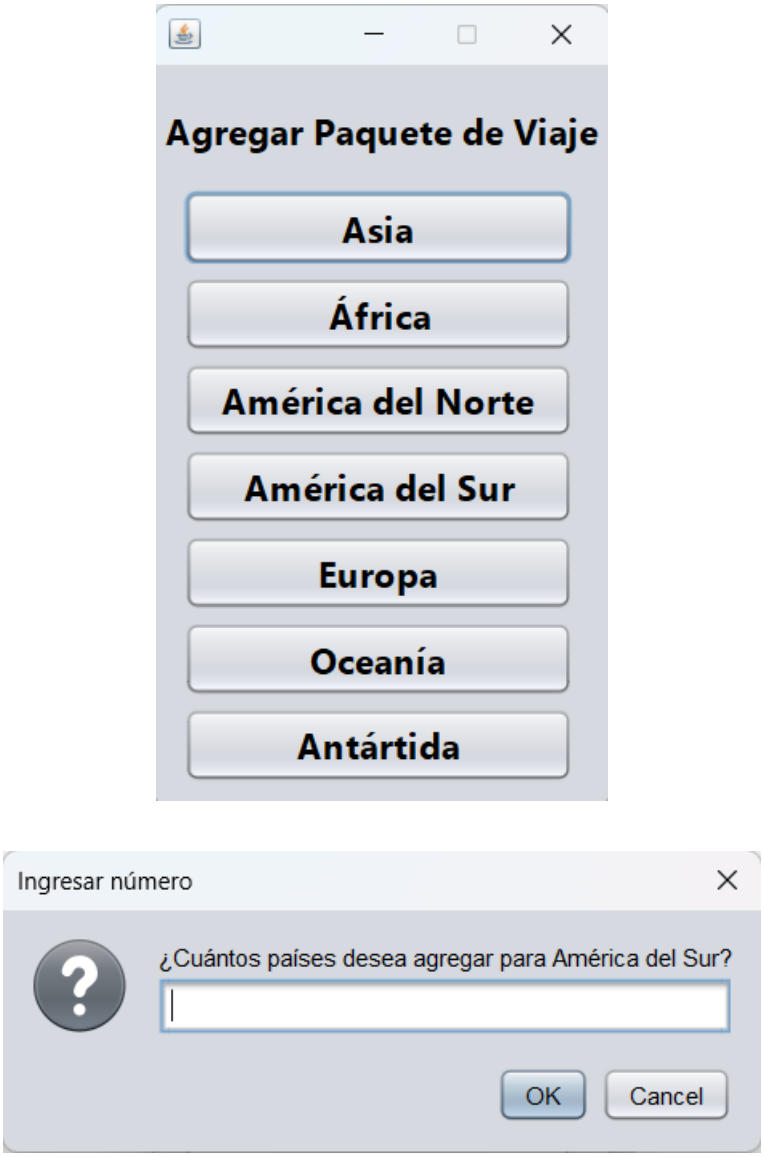


Figura 2. Ingresar un Nuevo Paquete.

Nota: al agregar un nuevo paquete, se le solicitarán una gran cantidad de atributos. En caso de que presione "Salir" (X) o "Cancelar", lo que haya ingresado hasta ese momento no se guardará. Por ejemplo, si llega al punto de agregar un CityTour a una ciudad y realiza la acción mencionada anteriormente, solo se guardarán los atributos de la ciudad, y lo ingresado en el CityTour se perderá.

Referencias

Capdevila, A. (s. f.). El patrón compuesto (Composite) en C#. https://albertcapdevila.net/patron-compuesto-composite-csharp/

Composite. (s. f.). https://refactoring.guru/es/design-patterns/composite

Composite | Marco de Desarrollo de la Junta de Andalucía. (s. f.). https://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/184

Equipo editorial de IONOS. (2020, 9 noviembre). El patrón composite: ejemplos de soluciones para jerarquías parte-todo. IONOS Digital Guide. https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/patron-composite/

Gala, F. J. G. (2015, 24 enero). Qué es el patron de diseño Composite. Urban Tecno. https://www.mundodeportivo.com/urbantecno/android/patron-composite

Leiva, A. (2023, 27 junio). Composite – patrones de diseño. DevExpert. https://devexpert.io/composite-patrones-diseno/

Oblancarte. (2020, 12 julio). Patrón de diseño - Composite - Oscar Blancarte - Software Architecture. Oscar Blancarte - Software Architecture. https://www.oscarblancarteblog.com/2014/10/07/patron-de-diseno-composite/

Patrón de diseño composite (estructural). (s. f.). https://informaticapc.com/patrones-de-diseno/composite.php