



UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERIA ELECTRÓNICA Y TELECOMUNICACIONES
PROGRAMA DE INGENIERÍA DE SISTEMAS
Laboratorio de Ingeniería de Software 2

PRACTICA DE LABORATORIO No. 6

Patrón estructural Composite

OBJETIVO

- Esta práctica tiene por objetivo conocer, entender, explorar y usar el patrón **Composite** aplicado a nuestro caso de estudio. Es importante además comprender cuándo y cómo aplicarlo entendiendo sus ventajas y desventajas frente a otros patrones estructurales.

INTRODUCCIÓN

Nuestra agencia de viajes desea organizar paquetes continentales, hoy los turistas quieren conocer un continente completo Europa, Norteamérica, Oriente medio u Oceanía, son largas vacaciones de dos semanas o más en las que los viajeros viven experiencias únicas y conocen nuevas culturas.

Cada continente está compuesto por varios países, y a su vez cada país tiene varias ciudades a visitar y en ellas hay más de un lugar por conocer.

La Copa América de 2019 a celebrarse en Brasil es un ejemplo de este tipo de viajes y la siguiente lo será aún más ya que vinculará dos países Argentina y Colombia.

Estructurando paquetes continentales

Con el propósito de estructurar mejor aún nuestra plataforma, la empresa ha decidido establecer una estructura jerárquica para construir paquetes continentales seleccionables por los turistas. Cada paquete continental estará compuesto de paquetes de países y a su vez cada paquete de país estará compuesto de paquetes de ciudades (hojas). En cada ciudad el paquete se construirá con un vuelo de llegada, un hotel, un city_tour y plan de alimentación, además de las fechas de *check in* y *check out* en cada destino.

De esta forma, será muy fácil flexibilizar la creación de paquetes personalizados tal y como lo hemos empezado a desarrollar en nuestro proyecto desde su creación. Un turista escogerá inicialmente el continente a visitar, luego dentro de este estará en disposición de organizar su periplo seleccionando tantos países como desee y su economía lo permita, a la vez en cada país seleccionará una o varias ciudades y, ya en la ciudad, definirá lo que desea hacer (ver Figura 1).

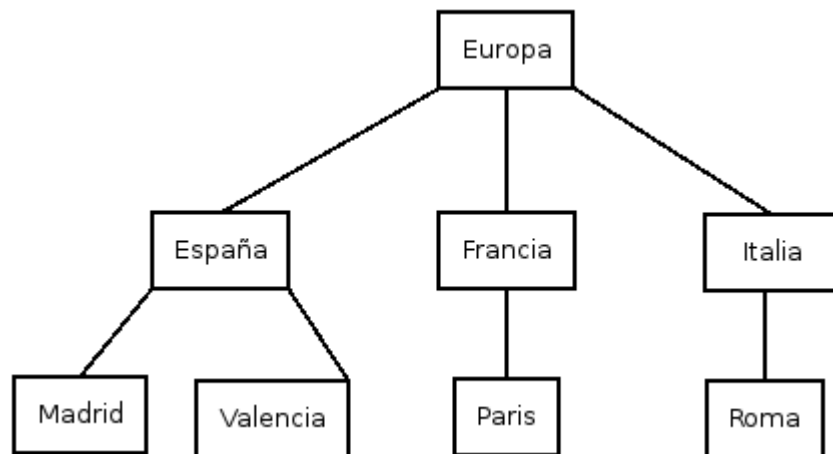


Figura 1. Jerarquía de componentes del paquete turístico

Revise las prácticas anteriores para que incluya todas las definiciones y conceptos desarrollados hasta el momento y utilice el patrón **composite** debidamente.

Cada ciudad en este caso puede asimilarse al destino de la práctica anterior.

Un *city_tour* es un corto paseo dentro de un destino o ciudad, con el propósito de conocer algún lugar de especial valor histórico, natural o cultural para vivir una experiencia memorable.

El diagrama de Clases es el siguiente.

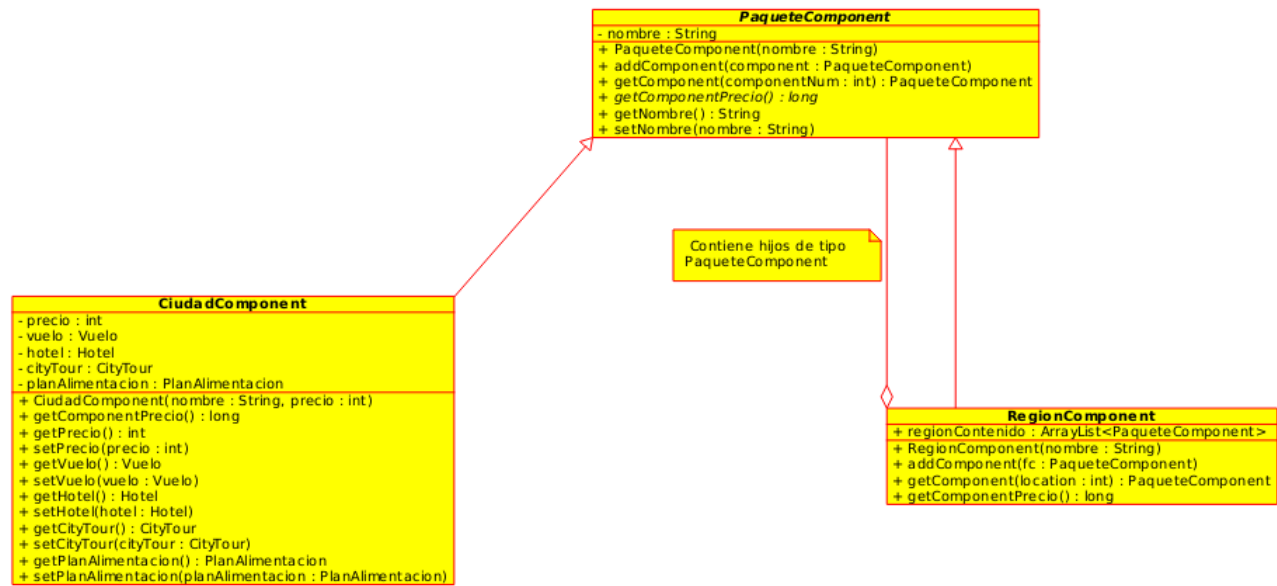


Figura 2. Diagrama de clases del composite

A continuación el código fuente de la Clase ClienteMain.

```

package co.agenciaviajes.cliente;

import co.agenciaviajes.negocio.CiudadComponent;
import co.agenciaviajes.negocio.RegionComponent;
import co.agenciaviajes.negocio.PaqueteComponent;

/**
 *
 * @author Libardo Pantoja, Ricardo Zambrano, Julio Hurtado
 */
public class ClienteMain {

    public static void main(String[] args) {
        // Ejemplo de un paquete a Europa, que visitará tres países: Italia, España y
        Francia
        PaqueteComponent principal = new RegionComponent("Europa");
        PaqueteComponent pais1 = new RegionComponent("Italia");
        PaqueteComponent pais2 = new RegionComponent("Francia");
        PaqueteComponent pais3 = new RegionComponent("España");

        PaqueteComponent ciudad1 = new CiudadComponent("Roma", 1000000);
        PaqueteComponent ciudad2 = new CiudadComponent("Paris", 2000000);
        PaqueteComponent ciudad3 = new CiudadComponent("Madrid", 3000000);
        PaqueteComponent ciudad4 = new CiudadComponent("Valencia", 2000000);
        principal.addComponent(pais1);
        principal.addComponent(pais2);
        principal.addComponent(pais3);

        pais1.addComponent(ciudad1);
    }
}

```

```
        pais2.addComponent(ciudad2);
        pais3.addComponent(ciudad3);
        pais3.addComponent(ciudad4);

        // El cliente se refiere tanto a los componentes compuestos, como a los
individuales, de manera uniforme
        System.out.println(" Región = "
            + principal.getNombre() + " " + principal.getComponentPrecio() + " Pesos");
        System.out.println(" Pais 1 = "
            + pais1.getNombre() + " " + pais1.getComponentPrecio() + " Pesos");
        System.out.println(" Pais 2 = "
            + pais1.getNombre() + " " + pais2.getComponentPrecio() + " Pesos");
        System.out.println(" Pais 3 = "
            + pais1.getNombre() + " " + pais3.getComponentPrecio() + " Pesos");

    });
}
```

La salida del programa debería ser algo como:

```
Región = Europa 8000000 Pesos
Pais 1 = Italia 1000000 Pesos
Pais 2 = Italia 2000000 Pesos
Pais 3 = Italia 5000000 Pesos
```

¿QUÉ SE DEBE ENTREGAR?

Se puede trabajar en grupos de dos personas.

1. Implementación en Java. Al docente debe entregarle la URL del repositorio en github de su trabajo.
2. Proponga en el cliente main otra composición para conocer paises y ciudades de tierra santa.
3. Implemente las pruebas unitarias.

TAREA

- Como tarea proponga una forma diferente de componer los paquetes turísticos (Similar a cómo lo hicieron en la teoría con el profesor Julio Hurtado).