



UNIVERSIDAD DE COSTA RICA

Escuela de Ciencias de la Computación e Informática

CI 0136 Diseño de Software

Prof. Mauricio Arroyo (mauricio.arroyo@ucr.ac.cr)

I semestre 2024

**Laboratorio: Structural Patterns
(13 punto relativos)**

Objetivo:

Que el estudiante tenga un acercamiento con los patrones Estructurales, de manera que los tenga como parte de sus herramientas cuando realice el diseño de un sistema.

Indicaciones generales:

- Para este laboratorio usted trabajará con un compañer@ de su grupo de proyecto. Es importante que ambos participen.
- Siga los pasos descritos en la siguiente sección.
- Al finalizar el laboratorio cada grupo debe subir un archivo zip con cada patrón funcionado.

Laboratorio

Decorator Pattern

1. Abra el código "Notifier.rar". Este código tiene una clase notifier similar a lo que discutimos en clase. Implemente el patrón de decorador para poder tener un Slack Notifier, Facebook notifier, y un SMS notifier. Para efectos del ejemplo imprimir un mensaje en la consola es suficiente, no tienen que implementar la lógica real para enviarlo por cada canal.
2. Modifique el main para llamar a cada notificador por separado.
3. Incluya en el main al menos dos combinaciones distintas de los decoradores.

Adapter Pattern

1. Abra el código "Fortnite.rar". La empresa Epic Games, dueña de fortnite ha realizado una alianza con Nintendo. En una nueva actualización el personaje Mario de Mario's Bros, será un nuevo personaje de Fortnite. Nintendo le dió una librería con la clase Mario a Epic Games, de la cual no se tiene el código fuente (en el laboratorio está pero asuma que modificarla no es una opción).
2. Les han pedido a ustedes hacer una prueba de concepto de manera que Mario pueda participar en el juego, a pesar de no ser uno de los ForniteCharacters. Todos los personajes del sistema heredan de ForniteCharacter. Como usted conoce el patrón de Adapter, impleméntelo de manera que podamos usar a Mario en Fornite.
3. El método attach está definido en cada clase, aunque el código podría haber sido movido a la clase base, el equipo de Epic Games decidió diseñarlo así para tener comportamientos diferentes en el ataque de cada personaje, que para efectos del lab no es importante.



Proxy Pattern

1. Abra el código "ImageProxy.rar"
2. Este código contiene la implementación simulada de una aplicación que carga imágenes de alta resolución para ser mostradas al usuario.
3. Nuestro cliente se está quejando de que las imágenes duran mucho en cargar (lo que se simula en este código con un sleep de 5 segundos por imagen).
4. Se explicó al cliente que la clase que carga las imágenes es de un tercero, del cual no tenemos el código (aunque en el lab está ☺) y que por lo tanto no podemos modificarlo y que el problema se da porque aun y cuando una imagen no es utilizada, la misma se carga al iniciar la aplicación, cuando se cargan todos los datos. (vea el método Main).
5. Un compañero de trabajo mencionó el patrón de Proxy, y específicamente un Virtual Proxy para mejorar el desempeño de la aplicación y su tarea es utilizar este patrón.
 - a. Cree una clase ImagenProxy que cargue la imagen únicamente cuando realmente se va a mostrar.
 - b. Descomente la prueba que su compañero de trabajo sugirió e implementó en el main, valide sus cambios con los datos arrojados con el cronometro.

MVC Pattern

Para Model View Controller, vamos a seguir un laboratorio elaborado por Microsoft, tiene un poco de todo, incluyendo trabajar con Base de datos, pero paso a paso.

Siga la guía en la siguiente dirección. Esto deben hacerlo en una computadora con Visual Studio.

<https://dotnet.microsoft.com/en-us/apps/aspnet/mvc>