

Structural Design Patterns

We will examine the following patterns:

- **Proxy** : A pattern that allows us to control access to another object
- **Decorator** : A common pattern to augment the behavior of an existing object dynamically
- **Adapter** : A pattern that allows us to access the functionality of an object using a different interface

The line between proxy and decorator

Decorator pattern is defined as a mechanism that allows us to enhance an existing object with new behavior, while the **Proxy pattern** is used to control access to a concrete or virtual object.

Conceptual difference between the two patterns.

Decorator pattern

- wrapper
 - You can take different types of objects and decide to wrap them with a decorator to enhance their capabilities with extra functionality.

Proxy pattern

- used to control the access to an object and it does not change the original interface. For this reason, once you have created a proxy instance, you can pass it over to a context that expects the original object.