

# Adapter

## Objective

It allows you to adapt or modify an existing interface in such a way that to otherwise incompatible classes can interact.

## Function

Adapt an interface so that it can be used by a class that otherwise I couldn't use it.

## Structure

This implementation uses the principle of object composition: the adapter implements the interface of one object and wraps around the other. It is can be implemented in all popular programming languages.

The structure that meets this pattern is shown in Figure 1

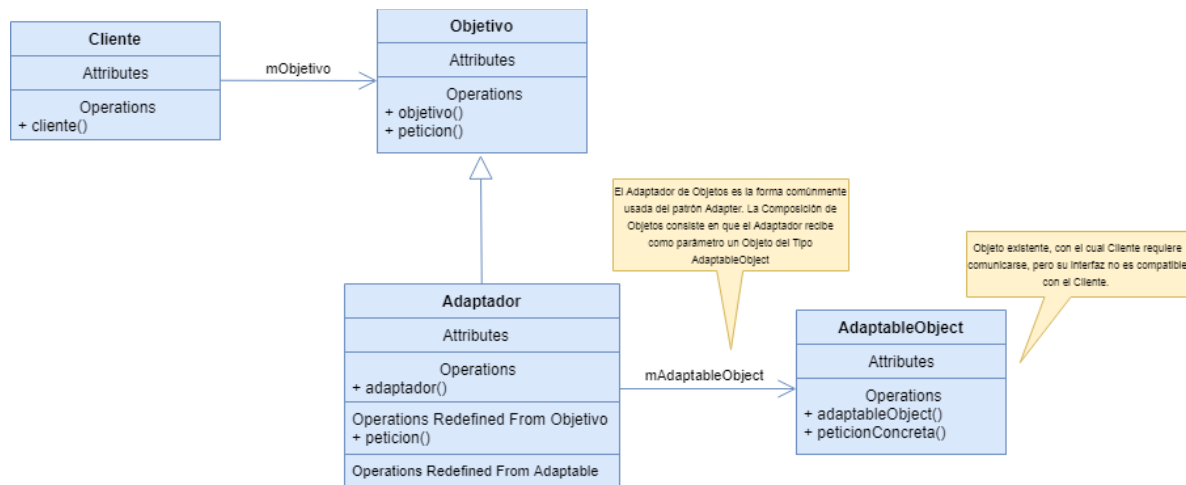


Figure 1: UML Diagram Adapter Pattern

## Applications

- It is required to create a reusable class, which cooperates with unrelated classes or not intended; that is, classes that do not necessarily have compatible interfaces.
- Different sets of methods need to be supported for different purposes.
- You need to use an existing class via an interface that is incompatible with that class.

- You want to allow the user to select between different GUIs (Graphical User Interfaces), to exploit the same system in the way that is most convenient for him user comfort.

## Design Patterns Collaborators

- No design patron is a collaborator.

## Scope of action

Applied at the object level.

## Problem

To make use of an object, whose interface is incompatible it is necessary to create a new one or otherwise modify the application so that can be integrated into that interface; in either case, the increase in complexity is remarkable.

## Solution

The Adapter design pattern allows adapting the original interface of an object with one that is compatible with the object expected by the client class; that is, the pattern is seen as a thin layer of code between two objects called "syntactically incompatible".

## Diagram or Implementation

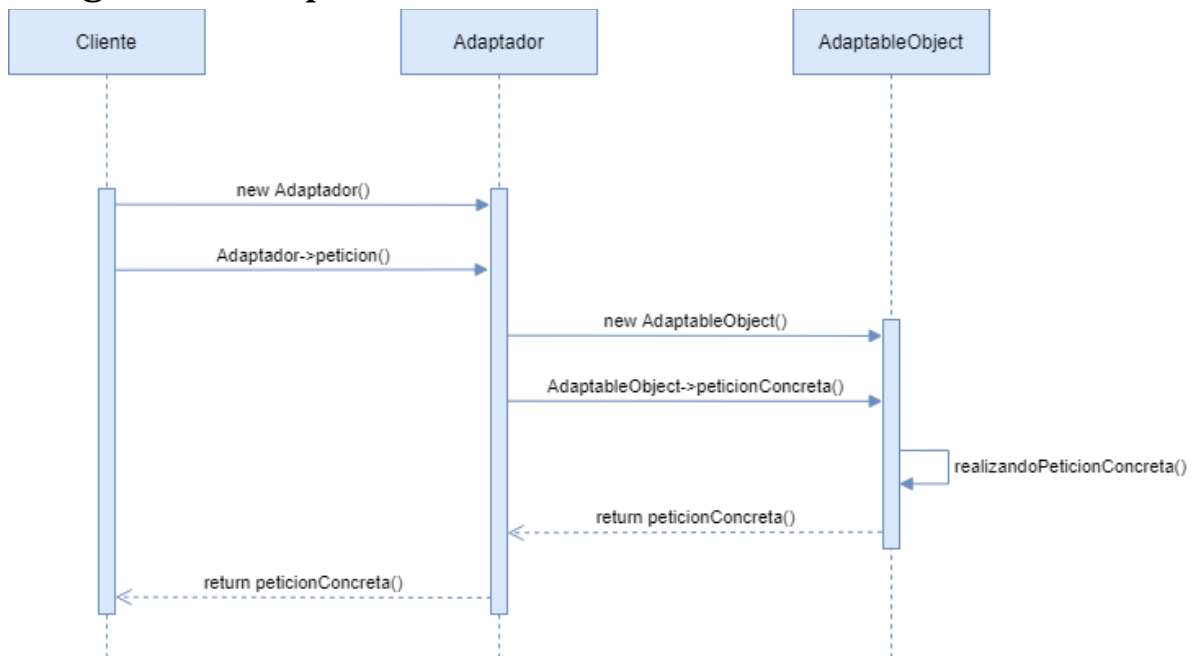


Figure 2: UML Diagram Adapter Pattern

The behaviour of the adapter pattern is explained below using the sequence diagram in figure 2

- The Client class invokes the Adapter class with generic parameters.
- Adapter class converts generic parameters into specific parameters of the Adaptee component.
- The Adapter class invokes Adaptee.
- Adaptee class responds.
- The Adapter class converts the Adaptee response to a generic response for the client class.
- The Adapter class responds to Client with a generic response.

## Study Cases

### Banking Portal System

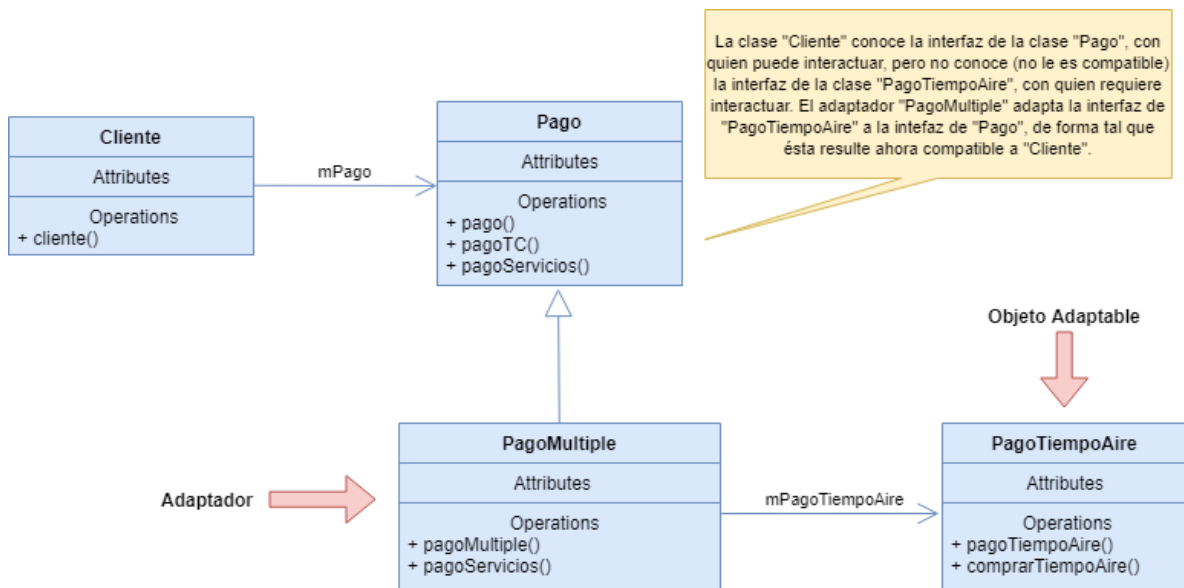


Figure 3: UML Diagram Banking Portal System

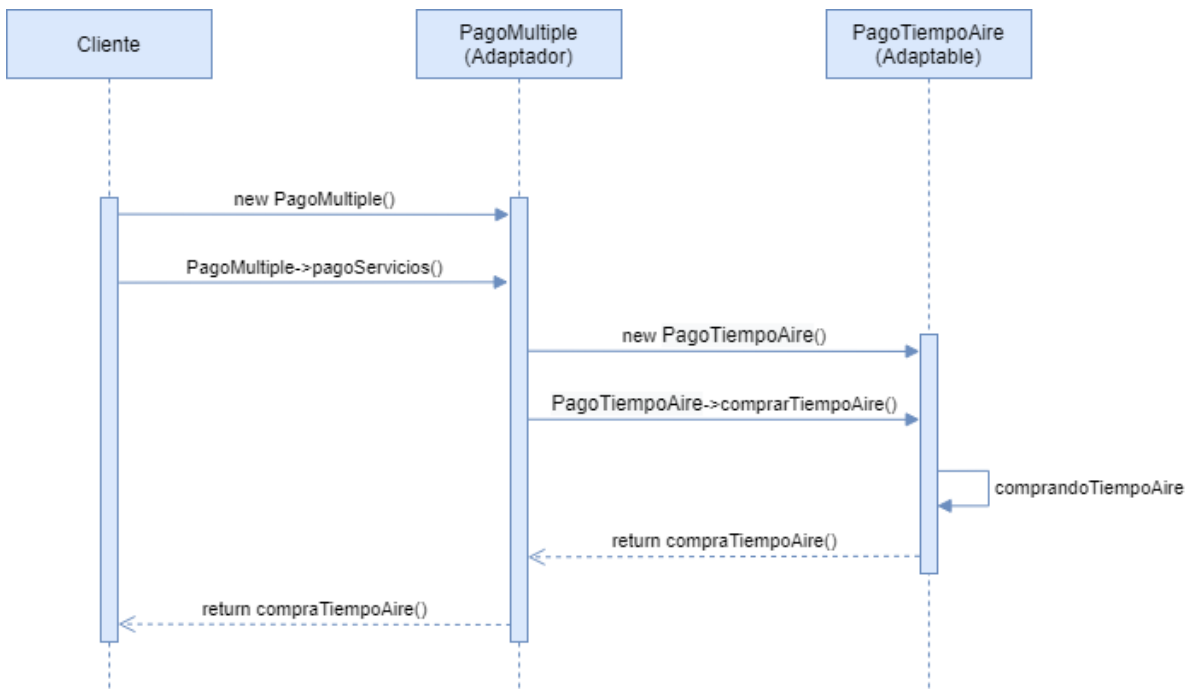


Figure 4: UML Diagram Banking Portal System

### Reservation and Sales System

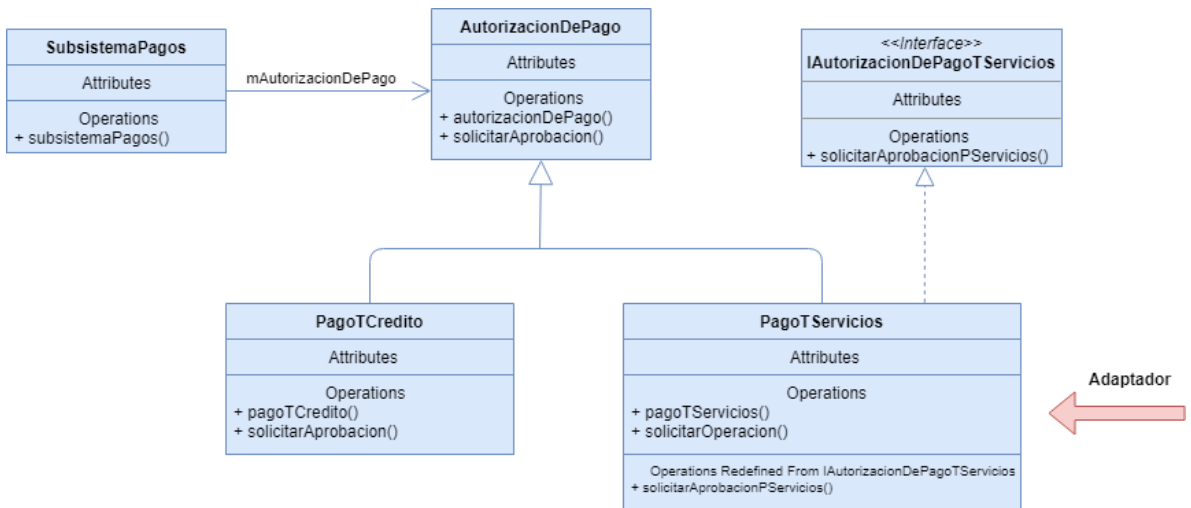


Figure 5: UML Diagram Reservation and Sales System

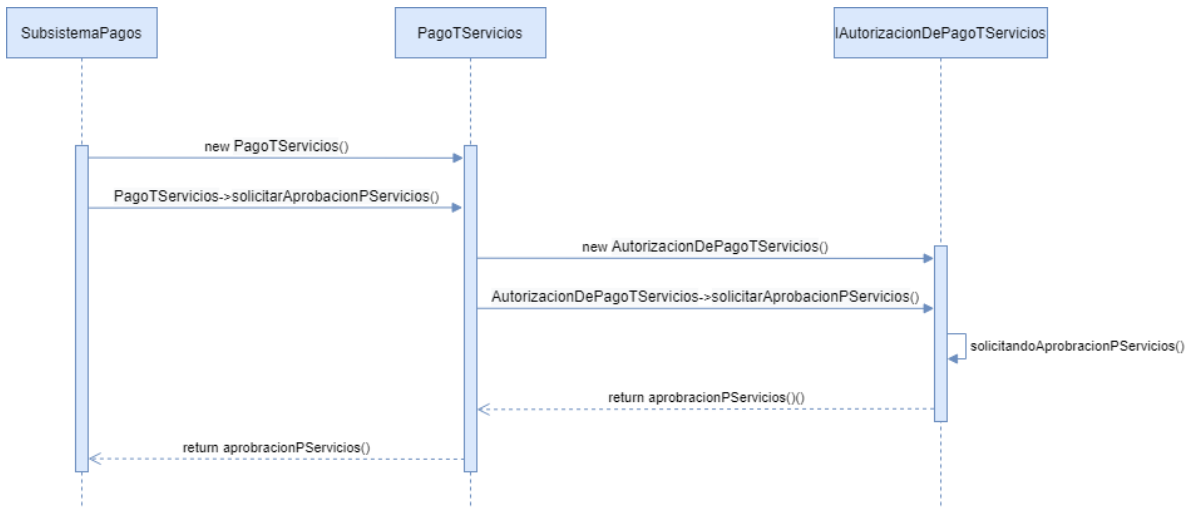


Figure 6: UML Diagram Reservation and Sales System