



## PRÁCTICA 4: A VECES LA MEJOR DECISIÓN ES NO TOMAR NINGUNA DECISIÓN Y ESO ES TAMBIÉN UNA DECISIÓN<sup>1</sup>

### 1. Introducción

Una técnica frecuentemente utilizada por los *patrones de diseño* para soportar variabilidad en el comportamiento dentro de un producto software es la de especificar la parte, o partes, variable de un método o clase como un método abstracto que tenga que ser sobrescrito en subclases representando las diferentes alternativas existentes. En este caso el método abstracto ejerce de punto de variación y las subclases serían las variantes. Ejemplos de dichos patrones son, entre otros, el patrón *Template Method*, el patrón *Strategy* o el patrón *State*.

Mediante esta técnica, la parte estable del comportamiento de estas clases se encapsula en una clase abstracta o interfaz, mientras que las partes variables queda confinadas dentro de diversas clases concretas. Idealmente, las clases que utilizan estas clases variables deberían depender sólo de la clase abstracta o interfaz creada, y no de las clases concretas que representan variantes concretas. Para conseguir este objetivo se utiliza normalmente una técnica conocida como *inversión de dependencias*.

El primer objetivo de esta práctica es que el alumno aprenda especificar comportamientos variables mediante la aplicación de un patrón de diseño concreto a un problema determinado. El segundo objetivo es que el alumno aprenda a *invertir dependencias* de manera que evite que las clases que dependen de interfaces o clases abstractas también dependan de sus clases concretas hijas. El tercer y último objetivo es que el alumno conozca alternativas a los patrones de diseño, como puede ser la utilización de funciones lambda, de manera que comprenda que muchos patrones de diseño existen simplemente por la complejidad accidental de ciertos lenguajes orientados a objetos.

El siguiente apartado refina esta serie de objetivos genéricos en un conjunto de objetivos concretos.

### 2. Objetivos

Los objetivos concretos de esta práctica son:

- (1) Aprender a aplicar el patrón *Strategy*.
- (2) Aprender a invertir dependencias.
- (3) Entender los problemas asociados a la inversión de dependencias.
- (4) Conocer y comprender la utilización de las funciones *lambda*.
- (5) Aprender a utilizar funciones lambda como alternativas a las estrategias.
- (6) Saber distinguir entre *complejidad esencial* y *complejidad accidental*.

Para alcanzar dichos objetivos, el alumno deberá aplicar el patrón *Strategy* y la técnica de inversión de dependencias a la situación que se describe a continuación.

---

<sup>1</sup> Frase pronunciada por D. Mariano Rajoy Brey (<https://goo.gl/D0j203>).



### 3. Visualización Internacional del Sistemas de Archivos *Sparrow*.

El Sistema de Archivos *Sparrow* debe, de acuerdo con su especificación, admitir en su nombre los caracteres propios del castellano. Estos caracteres son las vocales con tilde ortográfica, tanto en mayúsculas como en minúsculas, y nuestra querida letra eñe. No obstante, por otra parte, el Sistema de Archivos *Sparrow* debe ser compatible con dispositivos de visualización, como monitores e impresoras, que sólo soporten juegos de caracteres básicos y no incluyan por tantos algunos de estos caracteres especiales.

En estos último casos, la especificación oficial realizada por la organización *La Perla Negra* indica que existen tres alternativas diferentes para la visualización del sistema de archivos. La alternativa a utilizar dependerá tanto de las capacidades del dispositivo de salida que estemos utilizando como de las preferencias de cada usuario.

Si el dispositivo de salida soportase los caracteres propios del castellano, los nombres se muestran tal cual, sin necesidad de utilizar ninguna alternativa.

Si el dispositivo de salida soportase las vocales acentuadas, como la *á*, pero no soportase la letra *ñ*, existen dos alternativas disponibles a elección de cada usuario: (1) la *gallega*; y, (2) la *catalana*. En ambas alternativas, las vocales acentuadas ortográficamente permanecen sin modificar.

Cuando se usa la estrategia gallega, la letra *ñ* se reemplaza por los caracteres *nh*. Por ejemplo, *Briñón* se mostraría como *Brinhón*. En el caso de la estrategia catalana, la letra *ñ* se sustituye por la combinación de caracteres *ny*. Es decir, *Briñón* se visualizaría como *Brinyón*.

Si el dispositivo de salida no soportase ni la letra *ñ* ni las vocales con tilde ortográfica se debe utilizar entonces la opción *internacional*. En la opción internacional, en primer lugar, se sustituye cada vocal con tilde ortográfica por su correspondiente versión sin adornos fonéticos. A continuación, se reemplaza la letra *ñ* utilizando la estrategia catalana o la estrategia gallega, en función de las preferencias de cada usuario. Por tanto, en la práctica existen dos variantes de la alternativa internacional, la *internacional gallega* y la *internacional catalana*.

Utilizando la estrategia *internacional gallega*, *Briñón* se mostraría como *Brinhon*, mientras que de acuerdo con la estrategia *internacional catalana* el resultado sería *Brinyon*.

### 4. Actividades

El alumno, para alcanzar los objetivos planteados, deberá realizar satisfactoriamente las siguientes actividades:

1. Hacer que los visitantes de impresión creados en la práctica dedicada al patrón *Visitor* soporten, mediante la aplicación del patrón *Strategy*, las alternativas descritas en la sección anterior.



2. Invertir todas las dependencias que se hayan introducido como consecuencia de la aplicación del patrón *Strategy*.
3. Crear y ejecutar los casos de prueba necesarios para probar el correcto funcionamiento de las diferentes estrategias creadas.
4. En un proyecto aparte, hacer que los visitantes de impresión creados en la práctica dedicada al patrón *Visitor* soporten, mediante la utilización de funciones lambda, las alternativas descritas en la sección anterior.
5. Crear y ejecutar los casos de prueba necesarios para probar el correcto funcionamiento de la implementación creada basada en lambdas.

Los puntos 1 a 3 representan el mínimo a realizar para poder superar la asignatura, quedando los puntos 4 y 5 reservados para aquellos alumnos que quieran obtener calificaciones por encima del aprobado.

Para facilitar la realización de la práctica, el alumno tiene disponible en la plataforma Moodle dos clases, denominadas *ReemplazaCaracteres* y *OrdenaLambda*, implementadas en C#. Estas clases contienen código de ejemplo que puede servir de referencia y ayuda a la hora de implementar ciertos aspectos de la práctica que no se consideran relevantes para la consecución de los objetivos de la misma.

La clase *ReemplazaCaracteres* contiene la implementación de la función *desargentinizar* que sustituye la letra *y* por los caracteres *ll*. Puede utilizarse como base para implementar los métodos que sustituyen ciertas letras por otros patrones.

La clase *OrdenaLambda* contiene una implementación genérica del método de ordenación de la burbuja basado en funciones lambda. Se adjunta también un programa de prueba, contenido dentro de una clase *Runner*, que utiliza los métodos de la clase *OrdenaLambda*. Esta clase pretende servir de ejemplo de cómo se declaran y utilizan las funciones lambda en C#.

La siguiente sección describe los pasos que se deben seguir para aplicar el patrón *Strategy*.

## 5. Pasos para Aplicar el Patrón *Strategy*

Para aplicar el patrón *Strategy* se recomienda seguir los siguientes pasos:

1. Identificar el punto de variación del comportamiento donde se debe aplicar el patrón. En dicho punto sería necesario crear un *if* (o un *switch*) con multitud de ramas, normalmente una por cada variación del comportamiento existente. A este *if* lo denominaremos en adelante como el *if infernal*.
2. Situar en dicho punto de variación, es decir, donde debería estar situado una llamada a un método situado en una interfaz (o clase abstracta). Se recomienda utilizar como sufijo del nombre de dicha interfaz *Strategy*, para indicar que dicha clase representa una estrategia abstracta.
3. Por cada posible variante de la estrategia, es decir, por cada rama del *if infernal*, crear una subclase que herede de la interfaz *Strategy*. Sobrescribir el método abstracto con



la implementación que corresponda a esa variante concreta. Dicha implementación prácticamente la misma que la que estaría en la correspondiente rama del *if infernal*.

La siguiente sección contiene una serie de criterios cuyo objetivo es permitir al alumno autoevaluarse a *grosso* modo.

## 6. Criterios de Comprobación

Para verificar que el alumno ha implementado correctamente el patrón *Strategy*, se aconseja verificar los siguientes puntos:

1. Los visitantes de impresión no tienen bloques *ifs* con multitud de ramas a consecuencia de la incorporación de las estrategias gallega, catalana, internacional catalana e internacional gallega. En el caso ideal, los visitantes no tienen ningún bloque *ifs*, pudiéndose admitir como correctas implementaciones que tengan un *if* con, a lo sumo, dos ramas. Si alguien tiene dudas de eliminar completamente los bloques *ifs*, se le aconseja fijarse un momento en el título de esta práctica.
2. Es recomendable no tener la estrategia internacional duplicada para las variantes catalana y gallega, evitando siempre la creación de redundancias.
3. Ningún visitante debería hacer referencia a las implementaciones concretas de las estrategias. De hecho, los visitantes deberían importar sólo la interfaz que representa la estrategia abstracta.

Pablo Sánchez Barreiro