

## Práctica 04 - Patrón Statregy e IoD

**A veces la mejor decisión es no tomar ninguna decisión y eso es también una decisión <sup>[1]</sup>**

[1] Frase pronunciada por D. Mariano Rajoy Brey ([ver](#)).

### 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 variable de un método como un método abstracto que tenga que ser sobrescrito en diferentes subclases, cada una correspondiendo a una de las diferentes variabilidades 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, mientras que las partes variables queda confinadas dentro de diversas clases concretas.

Idealmente, las clases que utilizan las clases abstractas deberían depender sólo de la clase abstracta y no de las clases concretas que la implementan. Para conseguir este objetivo se puede utilizar una técnica conocida como *inversión de dependencias*.

El primer objetivo de esta práctica es que el alumno aprenda a especificar comportamientos variables mediante la aplicación del patrón *Strategy* 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 clases abstractas acaben también dependiendo de las clases concretas que la implementan. 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 ciertos patrones de diseño existen fundamentalmente 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.

## Objetivos

Los objetivos concretos de esta práctica son:

1. Aprender a aplicar el patrón *Strategy*.
2. Aprender a invertir dependencias.
3. Conocer y comprender el funcionamiento de las *expresiones lambda*.
4. Aprender a utilizar *expresiones lambda* sencillas.
5. Aprender a distinguir entre *complejidad esencial* y *complejidad accidental*.

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

## Visualización Internacional del *Sistema de Archivos Sparrow*

El *Sistema de Archivos Sparrow*, de acuerdo con su especificación original realizada por la organización *La Perla Negra*, debe admitir en su nombre los caracteres propios del castellano. Estos caracteres son las vocales con tilde ortográfica, como á, y tanto en mayúsculas como en minúsculas. Además, como no podía ser de otra forma, debe admitir nuestra querida letra ñ. Por otra lado, dicha especificación también establece que 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 tanto, algunos de estos caracteres especiales.

En estos casos, la especificación original 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 estrategia gallega*; y, (2) *la estrategia 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*.

Por último, si el dispositivo de salida no soportase ni la letra `ñ` ni las vocales con tilde ortográfica la especificación original del Sistema de Archivos Sparrow establece que se use la *estrategia internacional*. En esta estrategia, 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 o bien la estrategia catalana o bien la estrategia gallega, en función de las preferencias de cada usuario.

Por tanto, en la práctica existen dos variantes de la estrategia internacional, que son la *estrategia internacional gallega* y la *estrategia internacional catalana*. A modo de ejemplo, utilizando la *estrategia internacional gallega*, *Briñón* se mostraría como *Brinhon*, mientras que con la *estrategia internacional catalana* el resultado sería *Brinyon*.

## Actividades

El alumno, para alcanzar los objetivos de esta práctica, 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 estrategias de impresión para el *Sistema de Archivos Sparrow*.
2. Invertir todas las dependencias que se hayan introducido en los visitantes como consecuencia de la aplicación del *patrón Strategy*.
3. En un proyecto o carpeta aparte, hacer que los visitantes de impresión creados en la práctica dedicada al *patrón Visitor* soporten, mediante la utilización de *expresiones lambda*, las estrategias de impresión para el *Sistema de Archivos Sparrow*.

Para facilitar la realización de la práctica, se pone a disposición del alumno los siguientes ficheros de código fuente.

 [ReemplazaCaracteres.cs](#)

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 las estrategias de visualización del *Sistema de*

 [OrdenaLambda.cs](#) e  [InvocaLambda.cs](#)

La clase `OrdenaLambda` contiene una implementación genérica del método de ordenación de la burbuja que utiliza *expresiones lambda*. La clase `InvocaLambda` es un programa de pruebas que invoca al método de ordenación de la clase `OrdenaLambda`. El objetivo de estas clases es el de servir de ejemplo acerca de cómo se declaran y utilizan las *expresiones lambda* en C#.

## Criterios de Autoevaluación

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 subtítulo de esta práctica.
2. Ningún visitante debería tener referencias o crear implementaciones concretas de las estrategias implementadas. De hecho, los visitantes deberían importar sólo la interfaz que representa la estrategia abstracta.