



UNIVERSIDAD  
**NACIONAL**  
DE COLOMBIA

**Universidad Nacional de Colombia - sede Bogotá**  
**Facultad de Ingeniería**  
**Departamento de Sistemas e Industrial**  
**Curso: Ingeniería de Software 1 (2016701)**

**Grupo Passione**

**Jorge Alandete-jalandete@unal.edu.co Maicol Garzón-mgarzonm@unal.edu.co**  
**Santiago Maldonado-samaldonado@unal.edu.co Juan Medina-jumedinaca@unal.edu.co**

## Strategy

“El patrón de diseño de comportamiento que te permite definir una familia de algoritmos, colocar cada uno de ellos en una clase separada y hacer sus objetos intercambiables.”  
(online, s.f.)

El propósito de este patrón de diseño es separar comportamientos y algoritmos en distintas clases para un mismo objeto, logrando así que el actuar de la entidad pueda cambiar durante el tiempo de ejecución. También permite reforzar los principios SOLID aplicados a lo largo de todo el proyecto.

## Cómo fue implementado en el proyecto

Para la implementación del patrón de diseño Strategy en el proyecto se siguen una serie de pasos, primero se define una interfaz general de filtro, que declara un método de filtrado, encargada de comunicarse con los filtros creados, segundo por cada algoritmo de filtrado diferente que se desee implementar se crear una clase que implemente la interfaz de filtro..

De esta manera dependiendo del algoritmo de filtrado que se quiera aplicar, se va a crear una instancia de la clase del algoritmo para después ejecutar dicha instancia con el método de filtrado de la clase.

## Justificación de su uso en el contexto del sistema desarrollado

En la búsqueda de recetas, existen variedad de variables y atributos diferentes por los que se puede filtrar una receta para su búsqueda, como filtrar por ingredientes específicos de la receta, nombres, tiempo para su preparación, etiquetas, entre otras distintas formas. De esta manera la creación de cada forma de filtrar traería consigo nuevo código que implementar y adaptar al ya existente, lo que en un principio puede ser manejable, pero después de la implementación de varios filtros probablemente ocasionan un código espagueti cada vez más difícil de mantener. Así el uso del patrón de diseño Strategy mediante la implementación de los diferentes algoritmos de filtrado en clases hace el código escalable y mantenible en el tiempo.