|  |  |  |
| --- | --- | --- |
| CARACTERISTICAS | SOLUCION  CON  DECORATORS | SOLUCION  CON  STRATEGIES |
| Facilidad para agregar nuevos criterios de filtro | Implica crear una nueva clase que extienda a Filtro, y sobrescribir el método listarRecetasParaUnUsuario/1, aplicando el filtro correspondiente. | Implica crear una nueva clase que implemente Filtros, y sobrescribir el método test/2, que evalúa el criterio de filtro para el usuario y receta. |
| Posibilidad de eliminar filtros dinámicamente | Presenta una complejidad considerable, debido a la pérdida de identidad que sufren los objetos decorados. La solución no es trivial, y requiere de cierta lógica, que debería ser testeada. | Se puede realizar de manera muy sencilla, simplemente requiere remover el filtro en cuestión de la colección de filtros utilizada. |
| Posibilidad de intercambiar el orden en que se aplican los filtros o el posterior manejo de resultados | Al igual que a la hora de eliminar filtros, intercambiarlos se torna complejo.  Por otra parte, el post procesamiento también presenta cierta dificultad agregada, al tener que realizar controles para verificar que efectivamente se aplique luego de los filtros, y sea único. | El intercambio es relativamente sencillo, bastaría con ordenar la colección de filtros según se desee.  De igual manera, el post procesamiento no presenta dificultades, puesto que se aplica de manera secuencial, luego del filtrado, y como solo acepta un procesamiento por parámetro, la cantidad tampoco resulta un problema. |
| Simplicidad | Debido a los inconvenientes mencionados anteriormente, controlarlos todos implica una complejidad inicial considerable; luego, agregar nuevos filtros o procesadores resulta en un código simple. | La única complejidad que presenta es la reducción (foldeo) de la lista de filtros, aunque, no obstante, el código se mantiene simple. |
| Cohesión o cuántos objetivos resuelve el componente desarrollado | Presenta una mayor cohesión que la otra solución, puesto que mantiene el proceso de filtrado en el decorador que representa a cada filtro. Cada decorador filtra según su criterio, y delega el resto del filtrado al siguiente. | Es un poco menos cohesiva que la otra solución, pues el filtrado total es realizado en el repositorio de recetas, quien delega cada filtrado a la colección de filtros. |
| Mantenibilidad | La necesidad de más controles para cumplir la lógica de negocio implica más código y una mayor probabilidad de errores, y por consecuencia, más tests y un mayor esfuerzo mantenimiento. | Dada que el código de esta solución es más simple, resulta más fácil de mantener. |