Jose Manuel Fonseca Uribe

202122456

DPOO

[Patrón Iterator](https://github.com/kan01234/design-patterns/tree/master/iterator-pattern)

El proyecto es sobre un restaurante que muestra su menú. El menú puede ser almacenado en listas o arrays. Ahí es posible identificar un problema y es que el almacenamiento está limitado solo a 2 opciones, debería ser posible almacenar el menú en otro tipo de estructura. El proyecto tiene 7 clases, entre las cuales están productos concretos, iterators, iterators concretos y aggregates. El patrón iterator se usa generalmente para recorrer una colección sin dejar en exposición la estructura interna de esta. Este patrón permite acceder a los elementos de una colección de forma flexible, abstracta y separada del recorrido y la estructura de la colección. En el proyecto se puede ver el uso de iterator para recorrer el menú e imprimirlo sin importar si es lista o array. Al usar iterator en el proyecto se abstrae la estructura de datos y no es necesario preocuparse por el tipo de almacenamiento, pues será accedido a través de la interfaz iterator. Además, hay una separación de responsabilidades en la que en lugar de iterar directamente sobre la estructura de datos y acceder a los elementos, se delega esta tarea al iterator. Algunas desventajas que se pueden presentar al implementar iterator en el proyecto son que en el método printMenu, se pasa un objeto MenuIterator específico como argumento. El código depende directamente de las implementaciones concretas de los menús. Si se desea cambiar o agregar nuevas implementaciones de menús requeriría modificaciones en el código existente para adaptarse a esas nuevas implementaciones.

