

Proyecto 1

Raúl Nuño Valdés, 317204580

10 de noviembre de 2020

Justificación

Se usa el patrón **State** en la clase **Robot** ya que dependiendo de cual estado sea el que tenga va a ejecutar distintas acciones. Por ejemplo, si está **Suspendido** solo podrá activarse, o si está **Cocinando** solo podrá ir a cocinar el platillo pedido. Así se evita que el **Robot** ejecute métodos que no debería poder en cierto casos, como entregar un producto cuando ni siquiera tiene un orden.

Se usa el patrón **Adapter** para que la clase **Mercancia** pueda ser tratada como **MenuItem** pues tiene distintos parametros y métodos en comparación. Este es implementado en la clase **AdapterMenuItem**.

Se usa el patrón **Strategy** para cambiar la forma de cocinar un **MenuItem** ya que cada uno tiene distintos metodos de preparacion pero al final todos se cocinan. Se ve en **Cocina** y las clases que le implementan.

En estas mismas clases se usa la idea de **Facade** ya que lo único que tienen de similar es que se cocinan, pero, como ya se dijo, sus métodos son diferentes.

Se usa **Iterator** en **Menu** y las clases que le extiende pues cada una tiene una estructura de datos distinta y la forma más sencilla para recorrerlas sin tener que idear una forma exclusiva para cada una es usando un iterador. Se usa el mismo que tienen las estructuras **ArrayList**, **Collection** y **List** que es el de `java.util.Iterator`.

Además, el patrón **Composite** se usa tanto en los menus como en los items de estos para que sea más fácil recorrerlos y agruparlos.

Notas del proyecto

Se planeaba usar **Decorator** para **MenuItem**, pero como no se pide que el usuario sea capaz de ordenar cosas a agregar a su comida y también se dice que había que crear objetos ya definidos para correr el programa, se decidió no usarlo.

Notas para ejecutar

Si se ejecuta en CMD de Windows, es probable que aparezcan caracteres ANSI. Es porque se usaron colores para imprimir la información.

Si quiere ver los colores, use `reg add HKEY_CURRENT_USER/v VirtualTerminalLevel /t REG_DWORD /d 0x00000001 /f` en el CMD como administrador.

Pra revertirlo, use `reg add HKEY_CURRENT_USER/v VirtualTerminalLevel /t REG_DWORD /d 0x00000000 /f` en el CMD como administrador.

De esta pagina se obtuvo la información:

<https://www.codeproject.com/Tips/5255355/How-to-Put-Color-on-Windows-Console>

Ahora, para compilar se usa `javac -d bin -cp src src/main/Main.java` desde la carpeta `./proyecto`

Para ejecutar se usa `java -cp bin main/Main` también desde la carpeta `./proyecto`