

Proyecto 01

Modelado y programación

Objetivo: *El objetivo de este proyecto es resolver un problema propuesto considerando y valorando el uso de patrones de diseño de software. Diseña los diagramas UML* e implementa la solución para el siguiente problema usando los patrones de diseño vistos en clase que creas necesarios sin que tenga problemas de diseño como son Rigidez, Viscosidad, Fragilidad, Inmovilidad*

Problema:

El restaurante McBurguesas te ha pedido programar un robot que servirá como mesero/cajero y cocinero. Si el prototipo cumple con los requerimientos y las pruebas, los producirán en masa para toda su franquicia.

- Robot:
 - El robot debe estar encendido todo el tiempo esperando a que un cliente lo active, pero si no está trabajando puede estar suspendido.
 - El robot se encuentra en la barra donde atiende a los clientes. Al iniciar se encuentra suspendido, pero se activa cuando llega un cliente.
 - Una vez activado, el robot despliega todo el menú del restaurante.
 - Cuando el cliente elige lo que quiere comer, el robot deja la barra y se dirige al área de cocina. Una vez ahí, dependiendo del platillo que pidió el cliente, cambia su módulo de funcionamiento (explicado más abajo) y se pone a cocinar el platillo.
 - Al terminar de cocinar, camina hasta el cliente, le entrega el platillo al cliente y vuelve a suspenderse.
 - El robot pasa por el siguiente flujo: suspendido → atendiendo* → caminando → cocinando → caminando → atendiendo* → suspendido.
El flujo es automático. El robot siempre debe anunciar lo que está haciendo, pero no necesita autorización de nadie después de tener el pedido del cliente.
*se debe considerar que en este paso debe poder diferenciar entre “mostrarle el menú” y “entregarle la comida”.
- Menú: Inicialmente, existía un solo menú de hamburguesas, sin embargo el restaurante ha tenido tanto éxito que decidió incorporar varios menús con distinta comida rápida. El problema es que con cada nuevo menú ha habido un nuevo desarrollador que integra su propia forma de guardar un menú.
 - Implementa una clase MenuItem que guarde un nombre, una descripción, un indicador si el platillo es vegetariano, y un precio.
 - El primer menú será para **Hamburguesas**. Ese menú usará una **lista** para guardar sus elementos. La lista debe ser de objetos tipo MenuItem. Inventa 5

elementos de tu elección para guardar en este menú. Estos objetos serán fijos para todas las ejecuciones.

- El segundo menú será para **Pizzas**. Este menú usará un **arreglo** para guardar sus elementos. El arreglo debe ser de objetos tipo MenuItem. Inventa 4 elementos de tu elección para guardar en este menú. Estos objetos serán fijos para todas las ejecuciones.
- El tercer menú será de **Burritos**. Este menú usará una **hashtable** para guardar sus elementos. La hashtable debe ser de objetos tipo MenuItem. Inventa 4 elementos de tu elección para guardar en este menú. Estos objetos serán fijos para todas las ejecuciones.
- Aunque los 3 menús son disjuntos, el robot debe ser capaz de imprimirlos de forma unificada. El robot podrá acceder a todos los elementos de los 3 menús.

Hint: Los objetos de la clase HashTable tienen un método llamado values() que devuelve los valores en una estructura que extiende de Collection, y las estructuras de este tipo tienen un método llamado iterator().

- Módulo de funcionamiento: El robot cuenta con distintos funcionamientos de acuerdo a la tarea que debe realizar. Estos módulos están separados para darles mantenimiento como partes del robot. El robot internamente sólo necesita usar su instrucción programada para cocinar y el módulo determinará el platillo.
 - El robot sólo tiene un módulo a la vez. Debe cambiar su módulo antes de cocinar. Esta acción es automática y ocurre cuando el robot se encuentra cocinando, pero se ve reflejada en la pantalla del robot. El robot cambia su módulo de funcionamiento exclusivamente en la cocina.
 - Naturalmente, cada módulo le permite al robot trabajar de forma diferente. Las hamburguesas no se cocinan igual que las pizzas o los burritos, pero cada módulo le permite crear un platillo.
 - El primer módulo de funcionamiento incluye espátula, plancha, panes, lechuga, jitomates, carne, etc. Con este módulo es capaz de cocinar hamburguesas.
 - El segundo módulo de funcionamiento incluye horno, masa de pizza, salsa de tomate, queso, pepperoni, pimientos, champiñones, etc. Con este módulo es capaz de cocinar pizzas.
 - El tercer módulo de funcionamiento incluye comal, tortillas, frijoles, carne, vegetales, etc. Con este módulo es capaz de cocinar burritos.

- Debido al éxito del restaurante, el dueño ha decidido vender mercancía de la tienda (gorras, vasos, tazas, playeras, etc.). Deberás crear un objeto "Mercancía" que pueda representar estos artículos. Incluirá nombre, tipo de artículo, precio, color, material. Deberás crear al menos 10 objetos diferentes de tu elección.
- Al dueño le pareció buena idea integrar la venta de estos artículos en el robot, por lo que te pidió incluirlos en cualquiera de los 3 menús que ya existen(incluso puedes repartirlos en los 3 menús). Le da lo mismo en cuál lo hagas pero se opone a la idea de crear un cuarto menú. (Obviamente el jefe no es programador pero no puedes decirle que no.)

Hint: Claramente tendrás un problema para incluir objetos de tipo Mercancía en una estructura de datos que guarda MenuItem. ¿No había una forma de arreglar esto?

UML:

- Diagrama de clases (del sistema)
- Diagrama de estados (del robot)
- Diagrama de actividades (del flujo de trabajo)

Requerimientos:

- El programa debe empezar con un saludo del robot (que empieza suspendido). El primer menú le dará la opción de hacer un pedido o salir del programa.
- Una vez activado, el robot deberá atender al cliente, desplegará todo el menú y el usuario podrá ingresar lo que desea comer.
- El robot imprimirá en pantalla todo lo que ocurre. Debe imprimir su estado actual en todo momento y lo que está haciendo.
- Al entregarle el platillo al cliente, debe imprimir los detalles del platillo (nombre, descripción, si es vegetariano, y su precio) a manera de ticket.
- Finalmente el robot se suspenderá y el programa debe seguir corriendo, pues la única forma de terminar la simulación es con el menú inicial para salir del programa.
- El programa debe tratar a los platillos como objetos, por lo que no puede solamente imprimir lo esperado (estaría mal si sólo imprime una cadena fija).
- La mercancía debe estar integrada en uno o varios de los menús de comida que implementaste. La Mercancía NO debe ser un objeto de tipo MenuItem. La Mercancía NO debe heredar de MenuItem.

Evaluación:

Diagramas	2.5
Justificación de patrones	2.5
Implementación	2.5
Funcionamiento	2.5

Los lineamientos de entrega son los mismos que para las prácticas, excepto por los siguientes puntos:

- Este proyecto puede realizarse en equipos de 3 personas.
- La justificación de patrones reemplaza la parte teórica de la práctica. Debe incluirse en su archivo README.pdf