

# Modelado y Programación

## Proyecto 1

13 de Noviembre de 2020

### Integrantes:

- Gutiérrez Sánchez Claudia Itzel.
- Montiel Manriquez Ricardo.

No. de Cuenta: 112002433

No. de Cuenta: 314332662

### Justificación de Patrones:

Utilizamos 4 Patrones de Diseño:

#### Iterator:

Utilizamos el patrón Iterator para proveer una forma de acceder a los elementos de un objeto que los contiene de manera secuencial, sin exponer su representación interna. En este caso se crearon las clases IteradorHamburguesa, IteradorBurritos e IteradorPizzas, para poder recorrer las clases que contienen la lista de Hamburguesas, el hashtable de Burritos y el arreglo de Pizzas.

#### State:

Es un patrón de diseño de comportamiento que permite a un objeto alterar su comportamiento cuando su estado interno cambia. Parece como si el objeto cambiara subclase. Lo utilizamos para representar los estados del Robot (Suspendido, Atender, Caminar y Cocinar), así como las transiciones válidas que se lleven a cabo en ellos.

#### Strategy:

Este patrón es utilizado para poder identificar los aspectos que varían para poder encapsularlos. Lo utilizamos en los módulos de comportamiento de la cocina porque cambiamos de modulo dependiendo de la comida que se vaya a preparar para poder conseguir el proceso de preparación correcto.

#### Adapter:

Es un patrón de diseño estructural que permite la colaboración entre objetos con interfaces incompatibles. Lo ocupamos para poder agregar objetos de tipo mercancía en

una lista que solo contiene objetos de tipo menuItem. Con el adapter creamos un adaptador para que mercancía fuera “compatible” con menuItem.

### **Ejecución del Programa:**

Estando dentro de la carpeta ‘ src ’ abriremos la terminal y se ejecutara el comando ‘ javac Main.java ’ para compilar el programa. Después ejecutaremos el comando ‘ java Main ’ y con esto se mostrarla la simulación pedida.

### **Consideraciones:**

En el caso de la Mercancía tomamos en cuenta que no es necesario que vaya a ningún lugar para cocinar/conseguirla, porque ya está en la barra como en varios locales populares de comida rápida.

En el proyecto no se especificó si era necesario que se pudiera comprar más de un producto en cada pedido, por lo que en el caso de nuestro proyecto solo puedes comprar un producto por pedido.

En las clases de los Estados del Robot hay métodos vacíos, esto se debe a que se trata de transiciones ilegales en el programa, y también transiciones innecesarias. Pero debido a que implementan a una interfaz debíamos "implementar" todos los métodos de dicha interfaz.