

Práctica 0: Repaso de Java e introducción a Maven

Cómputo Concurrente 2024-2

Pérez Romero Natalia Abigail

February 4, 2024

1 Diagrama UML

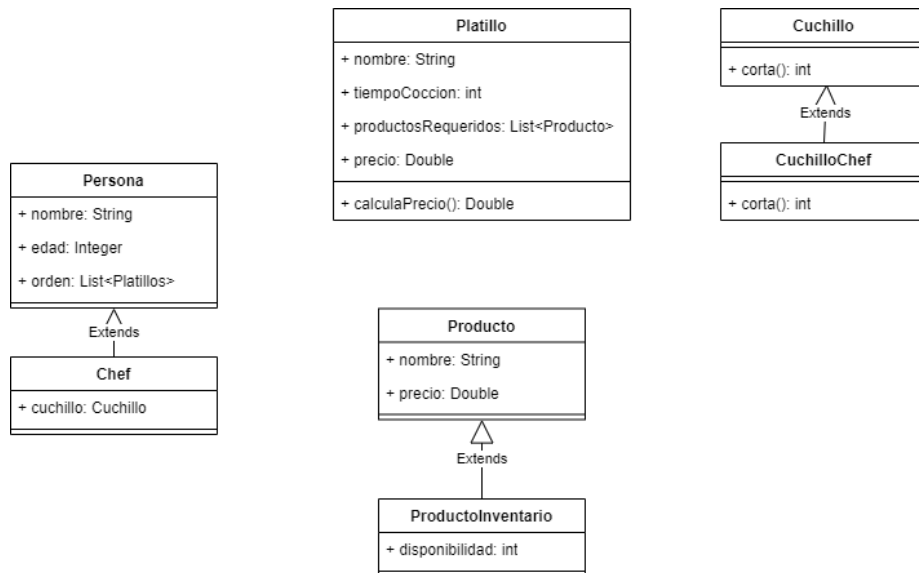


Figure 1: Diagrama UML

2 Justificación de Patrones de Diseño

Utilizo el patrón de diseño Strategy, en la clase Persona para permitir almacenar el pedido de una persona, de manera dinámica, es decir, que se pueda cambiar en tiempo de ejecución el algoritmo. Si se ordena un solo platillo, un producto

o varios platillos no seria necesario cambiar la clase, solo se cambia el algoritmo que se va a utilizar.

También en esta clase utilizo el patrón de diseño Adapter, para poder atender el pedido de un solo producto.

```
/**
 * Ordenar un platillo
 * @param platillo
 */
public void ordenar(Platillo platillo) {
    System.out.println( this.nombre + " ordena " + platillo.getNombre());
    this.orden.add(platillo);
}

/**
 * Ordenar un producto
 * @param producto
 */
public void ordenar(Producto producto) {
    System.out.println(this.nombre + " ordena " + producto.getNombre());
    this.orden.add(new Platillo(List.of(producto), producto.getPrecio(),
    tiempoCoccion:0, producto.getNombre()));
}

/**
 * Ordenar varios platillos
 * @param platillos
 */
public void ordenar(List<Platillo> platillos) {
    System.out.println(this.nombre + " ordena " + platillos);
    for (Platillo platillo : platillos) {
        this.orden.add(platillo);
    }
}
```

Figure 2: Diagrama UML

Luego en las clases Cuchillo-CuchilloChef, Persona-Chef, Producto-ProductoInventario podemos ver el patrón de diseño Decorator, ya que se pueden agregar atributos y métodos a las clases que implementan la interfaz Cuchillo o heredan de la clase Producto, Persona, que les permiten realizar otras operaciones como Chef que es capaz de cortar un producto o Persona que puede hacer un pedido.

3 Ejemplo de ejecución

4 Captura de Sonarlint

```

C:\Users\nataa\Documents\students_cc2024-2> c: && cd c:\Users\nataa\Documents\students_cc2024-2 && cmd /C ""C:\Program Files\Java\jdk-21\bin\java.exe" -XX:+ShowCodeDetailsInExceptionMessages -cp C:\Users\nataa\Documents\students_cc2024-2\Practica0\target\classes kass.concurrente.Main "
Iniciando simulacion
Creando inventario
Creando platillo
cliente 1 ordena Platillo 1
Chef Chef 1 Atendiendo a Cliente 1
Preparando platillo con cuchillo
Buscando Producto 1 en el inventario
Buscando Producto 2 en el inventario
Preparando Platillo 1 en 4 minutos
Platillo Platillo 1 listo
cliente 2 ordena Producto 3
Chef Chef 1 Atendiendo a Cliente 2
Preparando platillo con cuchillo
Buscando Producto 3 en el inventario
No se encontro Producto 3 en el inventario. Consiguiendo producto
Platillo Producto 3 listo
cliente 3 ordena Platillo 3
Chef Chef 1 Atendiendo a cliente 3
Preparando platillo con cuchillo
Buscando Producto 3 en el inventario
Preparando Platillo 3 en 5 minutos
Platillo Platillo 3 listo

```

Figure 3: Ejecución

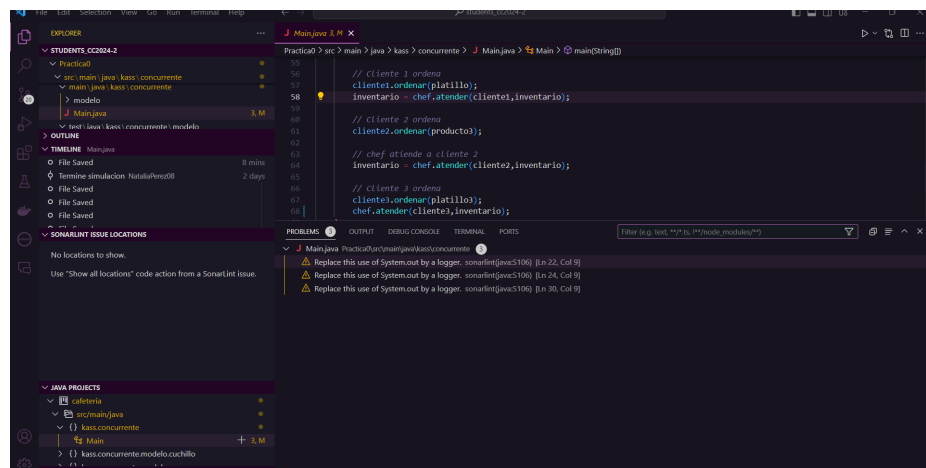


Figure 4: Sonarlint