Juan José Osorio (202021720)
Thais Tamaio (202022213)

Documento taller 1 – DPOO

A continuación se muestran los métodos agregados con su respectiva justificación:

Clase Restaurante:

1. getCombos(): Este método agregado no lo consideramos como necesario para el desarrollo del programa. No obstante, permite organizar de mejor manera el agregar un producto desde el menú. Por lo que se muestra una explícita diferencia entre seleccionar un producto y seleccionar un combo.

```
public ArrayList<Combo> getCombos()
{
    return this.combos;
}
```

2. getPedidos(): Decidimos agregar este método para realizar de manera más sencilla la comparación del pedido actual con todos los pedidos realizados con anterioridad. Por medio de este método se puede confirmar la existencia de un pedido idéntico.

```
public static ArrayList<Pedido> getPedidos()
{
    return Restaurante.pedidos;
}
```

Aplicación:

1. input(): Se agregó este método para poder obtener los parámetros necesitados por el programa.

Clase combo:

1. getProductos(): Por medio de este método se genera un Array List donde de guardan los productos que conforman un combo. Esto nos permite, acceder a esta información de manera más sencilla.

```java
public ArrayList<Producto> getProductos()
{
    return this.itemsCombo;
}
```

Clase producto ajustado:

1. addAgregado() and addEliminado(): Estos dos métodos permiten dejar de manera explícita el proceso de agregar y eliminar un ingrediente a su respectivo Array.

```java
public void addAgregado(Ingrediente nuevo)
{
    agregados.add(nuevo);
}

public void addEliminado(Ingrediente viejo)
{
    eliminados.add(viejo);
}
```

Implementación parte 3

Clase Combo:

1. getCalorias():

```java
@Override
public int getCalorias()
{
    int cantidadCalorias = 0;
    for (Producto p: this.itemsCombo)
    {
        cantidadCalorias += p.getCalorias();//Se obtiene la cantidad total de calorías.
    }

    return cantidadCalorias;
}
```

Este método fue implementado para obtener las calorías netas de un combo. Por lo que se suman las calorías de todos los productos que conforman un combo.

Clase Ingrediente:

1. getCalorias():

```java
public int getCalorias()
{
    return (this.calorias);//Obtener calorías.
}
```

Este método fue implementado para obtener las calorías de un ingrediente.

Clase Pedido:

1. equals(ArrayList<Pedido>):

```java
/**
 * Se comparan únicamente los pedidos realizados en una misma sesión.
 * @param ArrayList<Pedido> pedidos
 * @return boolean
 */
public boolean equals(ArrayList<Pedido> pedidos)
{


    for (Pedido p: pedidos)
    {
        if (this.itemsPedido.equals(p.itemsPedido))
        {
            return (true);
        }
    }

    return (false);

}
```

Este método fue implementado para comparar el pedido actual con todos los pedidos realizados en una misma sesión. De esta manera se puede encontrar si existen dos pedidos idénticos.

Clase ProductoAjustado:

1. getCalorias():

```
/**
 *Retorna las calorías totales del producto ajustado. Se agregan las calorías de los ingredientes agregados y vice versa.
 *@return int: Calorías
 */
@Override
public int getCalorias()
{
    int caloriasBase = this.base.getCalorias();

    for (Ingrediente i : this.agregados)
    {
        caloriasBase += i.getCalorias(); //Se suman las calorías de los ingredientes agregados.
    }

    for (Ingrediente i : this.eliminados)
    {
        caloriasBase -= i.getCalorias(); //Se restan las calorías de los ingredientes eliminados.
    }

    return caloriasBase; //Se obtienen las calorías totales.
}
```

Se suman las calorías a un producto cuando se agrega un ingrediente y se restan las calorías cuando se elimina un ingrediente. Por lo tanto, se obtienen las calorías netas del producto ajustado.

Clase ProductoMenu:

1. getCalorias():

```
@Override
public int getCalorias()
{
    return (this.calorias);//Se obtienen las calorías.
}
```

Se obtienen las calorías de un producto del menú.

Clase Restaurante:

1. getBebidas():

```
public ArrayList<Producto> getBebidas() //Se obtienen las bebidas.
{
    return this.bebidas;
}
```

Se obtiene la información de las bebidas, que ahora se encuentran en un archivo aparte.