

PROYECTO MÁQUINA BEBIDAS



🖥 Equipo: Javier San Juan Muela y Adrián Rangel Dorrego 🖥

► **ÍNDICE** ◀

► ÍNDICE ◀.....	2
ANÁLISIS	3
DEFINICIÓN.....	3
ESTRUCTURA.....	4
1. CLASES.....	4
2. ATRIBUTOS.....	4
3. MÉTODOS.....	5

ANÁLISIS



El **análisis** sobre este proyecto se **basará** en 2 partes:

- **Definición:** En esta apartado se explicará el proyecto de manera inicial y en qué consiste.
- **Estructura:** Después de la definición se expondrán las **clases**, **atributos** y **métodos** de manera sistemática, visual y comprensible que se van a usar en el proyecto.

DEFINICIÓN

Este proyecto consiste en una aplicación capaz de manejar y gestionar el motor lógico de una máquina de bebidas calientes (Café, Té y Chocolate).

De esta manera, esta aplicación va a ser capaz de realizar las siguientes funciones:

- Gestión del inventario de ingredientes y utensilios.
- Validación de pedidos.
- Contabilización de las ventas.

ESTRUCTURA

Una vez entendido en qué consiste este proyecto, lo posterior es entender cómo está compuesta cada una de sus funciones. Por ende, sus clases, atributos y métodos están compuestos de la siguiente manera:

1. CLASES

Las clases a definir dentro del proyecto son las siguientes:

→ AppMaquina.class

Esta clase va a ser la que va a **ejecutar** el programa, y, por ende, va a ser la **principal**. Sin esta clase no se va a poder ejecutar el programa.

→ Bebida.class

Esta clase va a definir las **bebidas**, y al definir el mismo, se van a usar las otras clases para establecer los **ingredientes** y **cantidades**

→ Ingrediente.class

Esta clase va a definir y gestionar los **ingredientes** de cada bebida

→ Utensilio.class

Esta clase va a definir y gestionar los **utensilios** usados para las bebidas.

→ Peso.class

Esta clase define el **tipo** de peso y la **cantidad** de cada **ingrediente**. Se diseñó de esta manera porque, al no existir una lista de clases para cada bebida, es **necesario** especificar en este objeto el tipo de peso y la cantidad correspondiente.

2. ATRIBUTOS

En base a las clases anteriormente definidas, los atributos para cada una serían:

→ AppMaquina.class

Dentro de esta clase, se definirá un atributo llamado **opción** (*int opcion*) para la elección del método que se ejecutará.

→ Bebida.class

Dentro de esta clase, se definirá la siguiente lista de atributos:

- *int cantidad* -> Este atributo define la cantidad de la bebida.

- `static int cantidadTotalBebidas` -> Este atributo define la cantidad de la bebida.
- `String nombre` -> Este atributo guarda el nombre de la bebida
- `Ingrediente[] ingredientes` -> En este atributo se definirá el objeto con los ingredientes que va a usar.
- `Utensilio[] utensilio` -> En este atributo se definirá el objeto que dispondrá información de qué utensilios se usarán para esta bebida.

→ Ingrediente.class

Dentro de esta clase, se definirá la siguiente lista de atributos:

- `Peso stock` -> En este atributo se guarda la cantidad de cantidad de cualquier ingrediente que hay disponible.
- `String nombre` -> En este atributo se definirá el nombre del ingrediente (ejemplo: azucar, leche, gramosCafé).

→ Utensilio.class

Dentro de esta clase, se definirá la siguiente lista de atributos:

- `int stock` -> En este atributo se define la cantidad disponible del utensilio (ejemplo: 200 -> vaso)
- `String nombre` -> En este atributo se define el nombre del utensilio

→ Peso.class

Dentro de esta clase se definirá la siguiente lista de atributos:

- `float cantidad` -> En este atributo se define la cantidad (ejemplo: 7 gramos de X).
- `String medida` -> En este atributo se define la medida del peso (ejemplo: 10 kg de X).

3. MÉTODOS

Para las clases que se han definido anteriormente podemos introducir los siguientes métodos para cada clase:

→ Peso.class

- `public float getCantidad():` Obtenemos la cantidad actual.
- `public float setCantidad(float nuevaCantidad):` Permite actualizar el peso.

→ Utensilio.class

- public boolean `hayStock`(int cantidad)
- public void `consumir`(int cantidad)
- public void `rellenar`(int cantidad)
- public int `getCantidad`(): Obtenemos la cantidad actual.
- public int `setCantidad`(int cantidad): Establecemos una cantidad.

→ Ingrediente.class

- public boolean `hayStock`(float cantidad)
- public void `consumir`(float cantidad)
- public void `rellenar`(float cantidad)
- public float `getCantidad`(): Obtenemos la cantidad actual.
- public float `setCantidad`(float peso): Establecemos una cantidad.

→ Bebida.class

- public boolean `verificarDisponibilidad`()
- public void `servir`()
- public int `getCantidad`(): Obtenemos la cantidad actual.
- public int `setCantidad`(int cantidad): Establecemos una cantidad.

→ AppMaquina.class

- public void `iniciar`(): Método principal que arranca el bucle del programa.
- public void `mostrarMenu`(): Imprime en consola las opciones disponibles.
- public void `procesarOpcion`(int opcion)
- public void `prepararBebida`(Bebida bebida)
- public void `mostrarInventario`(): Muestra todos los recursos.