

<b>Nombre:</b>	
<b>Código:</b>	<b>Fecha: 28 de Septiembre de 2018</b>

**Unidad 1: Introducción a la solución de problemas utilizando algoritmos**

- OE1.5. Utilizar expresiones simples: declaraciones de variables, asignaciones e invocaciones a métodos. Esto implica entender los conceptos de parámetro y de creación de objetos.
- OE1.6. Utilizar un ambiente de desarrollo de programas (incluyendo la compilación y ejecución de programas desde consola) y un espacio de trabajo predefinido, para construir la solución de un problema.
- OE1.7. Interpretar la documentación (modelos conceptuales y especificación de las operaciones) de una solución con el fin de: a) llevar a cabo la implementación con base en el diseño (rol del programador vs rol del diseñador), b) comprender el desarrollo en su totalidad, c) verificar la coherencia entre análisis, diseño e implementación (trazabilidad de los elementos en los artefactos del desarrollo) y, d) validar la solución propuesta en el diseño con las necesidades del usuario expuestas en los requerimientos funcionales.
- OE1.8. Aplicar el concepto de encapsulamiento en el modelamiento e implementación de soluciones bajo el paradigma de programación orientado a objetos.
- OE1.10. Declarar y utilizar variables diferenciando entre tipos de datos primitivos y tipos de datos referenciados.
- OE1.11. Construir las clases que implementan el modelo de la solución del problema.
- OE1.12. Construir las clases que implementan una interfaz de usuario (consola) e integrarlas con las clases que implementan el modelo de la solución del problema.
- OE1.13. Utilizar una arquitectura para un programa que permita repartir de manera adecuada las responsabilidades entre la interfaz de usuario y el modelo de la solución. El estudiante deberá poder explicar la importancia de mantener separadas las clases de estos dos dominios.

**Unidad 2: Solución de problemas utilizando estructuras de control condicionales**

- OE2.2. Utilizar expresiones aritméticas, lógicas, relacionales y operaciones con cadenas en el cuerpo de un método.
- OE2.3. Declarar y hacer llamados a métodos constructores declarados explícitamente en la misma clase y utilizar null en caso de que un objeto no haya sido inicializado.
- OE2.4. Utilizar las instrucciones condicionales como parte del cuerpo de un algoritmo, para poder considerar distintos casos de la solución de un problema.
- OE2.6. Interpretar los errores producidos en tiempo de ejecución para el caso en que se realicen llamados u operaciones con objetos que no han sido construidos.
- OE2.8. Valorar la coherencia estricta entre el diseño propuesto en el diagrama de clases y la implementación en el lenguaje de programación.

**Enunciado**

Juan Balde, una de las principales franquicias de tiendas de ventas de bebida a base de café quiere empezar un proyecto para poner una nueva sucursal en la Universidad Icesi. El líder de la franquicia se ha acercado a ti porque quiere construir un programa que le permita saber cuál de sus bebidas es la que le genera mayor ganancia neta y poder agregar nuevos productos a la tienda. Cada producto tiene un nombre, un tipo que puede ser (Bebida Caliente, Bebida Tibia y Bebida fría), un precio de venta, un porcentaje de ganancia neta<sup>1</sup> y la cantidad de unidades vendidas. En caso de que existan dos bebidas que tengan la misma cantidad de ganancia neta generada el algoritmo retornará aquella que tenga mayor cantidad de unidades vendidas, si persiste la igualdad puede retornar cualquiera de las bebidas. Recuerda que la ganancia neta de un producto se obtiene con la siguiente ecuación:

$$\text{Ganancia Neta Bebida} = \text{cantidad bebidas vendidas} * \text{precio de venta} * \text{porcentaje de ganancia neta}$$

De la tienda se conoce el nombre de la franquicia. Actualmente Cuenta con 2 bebidas pero espera en un futuro muy cercano tener hasta 4 bebidas. Se te hace entrega de la fase de análisis y de diseño las cuales puedes encontrar a continuación:

<sup>1</sup> El porcentaje de ganancia neta es un número Real entre 0 y 1 que indica qué porcentaje del precio de venta de un producto es ganancia neta o utilidad para un establecimiento.

## Requerimientos Funcionales

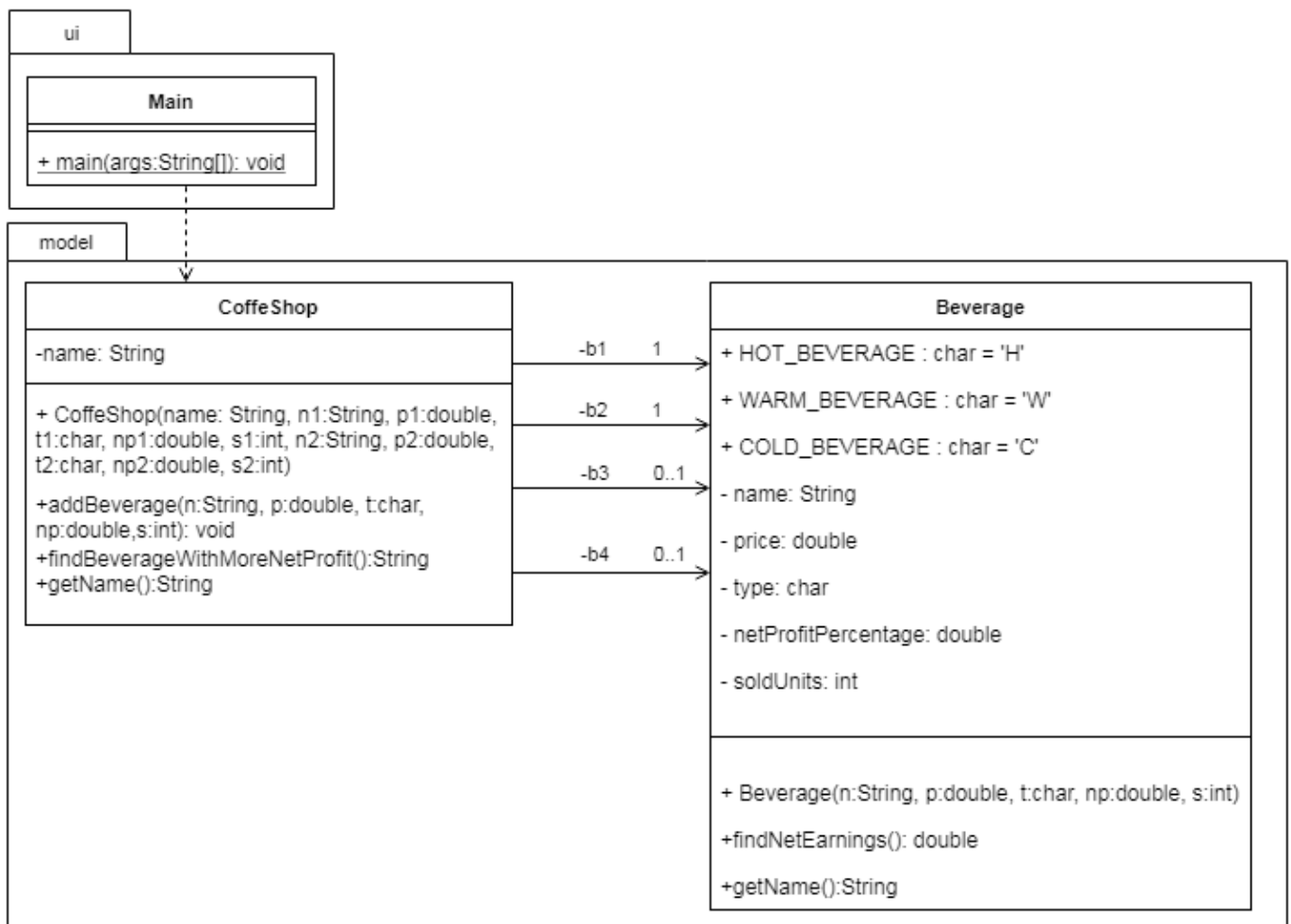
El programa debe estar en la capacidad de:

<b>Nombre:</b>	R1. Crear una Bebida
<b>Resumen:</b>	Permite registrar en el sistema una nueva bebida. Con su nombre, tipo, precio de venta, porcentaje ganancia neta y cantidad de unidades vendidas.
<b>Entradas:</b>	Nombre: String Precio venta: double Tipo: char Porcentaje ganancia neta: double Cantidad unidades vendidas: int
<b>Resultado:</b>	Se ha creado una nueva bebida si los datos son correctos.

<b>Nombre:</b>	R2. Encontrar la bebida con que generó la mayor ganancia neta.
<b>Resumen:</b>	Verifica la ganancia neta de cada una de las bebidas y retorna la mayor. En caso de que existan dos bebidas que tengan la misma cantidad de ganancia neta generada el algoritmo retornará aquella que tenga mayor cantidad de unidades vendidas, si persiste la igualdad puede retornar cualquiera de las bebidas.
<b>Entradas:</b>	<i>ninguna</i>
<b>Resultado:</b>	Se muestra el nombre de la bebida con mayor ganancia neta y el valor de su ganancia neta.

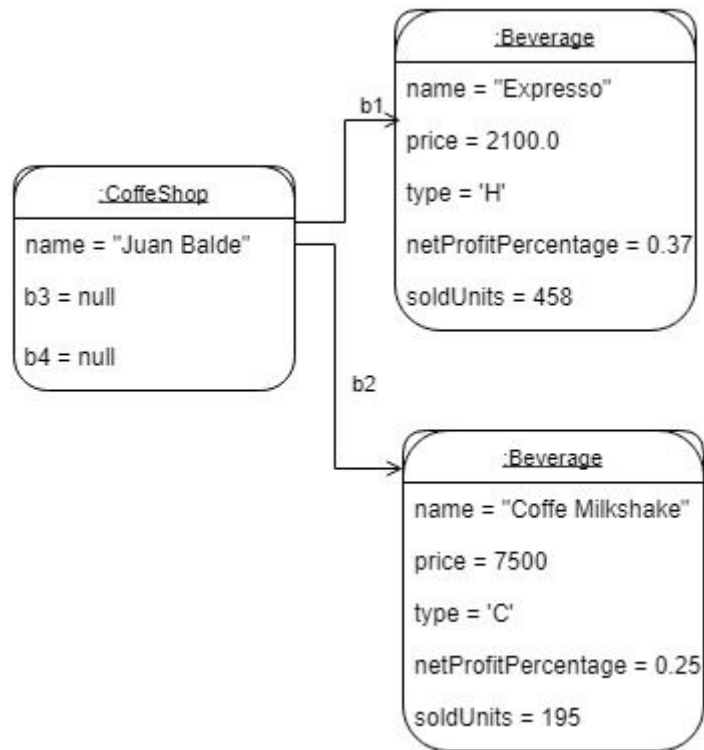
## Diagrama de Clases

El diagrama de clases se presenta a continuación:



### Diagrama de Objetos

Un posible diagrama de objetos con base en el diagrama de clases anterior se presenta a continuación:



### Entrega

Usted debe entregar un archivo en formato .zip cuyo nombre debe tener el formato PRIMERAPELLIDO\_PRIMERNOMBRE.zip en el cual se encuentre una carpeta con el nombre API\_PRACTICO1\_PRIMERAPELLIDO\_PRIMERNOMBRE, dentro de la cual se encuentren las carpetas de los respectivos paquetes definidos en el diagrama de clases y éstas a su vez contengan los archivos de las clases implementadas en Java.

En el método main usted debe implementar un escenario a través de la creación de los objetos con los valores predeterminados. Una vez realizado ésto el programa deberá ejecutar el requerimiento 1 el cual implica leer por consola los valores que considere necesarios, finalmente debe imprimir en consola el resultado respectivo del requerimiento 2 y terminar su programa.