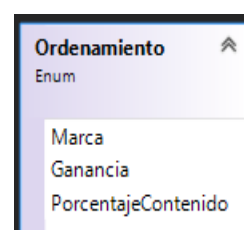
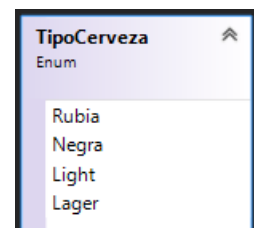
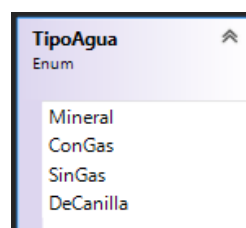
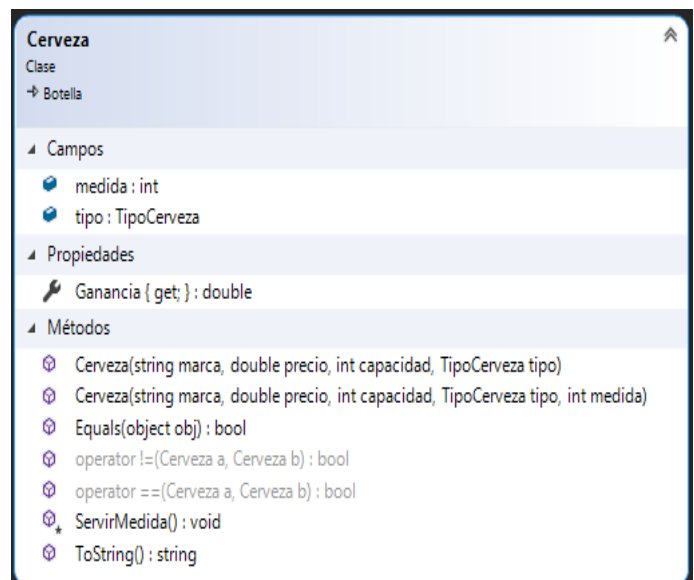
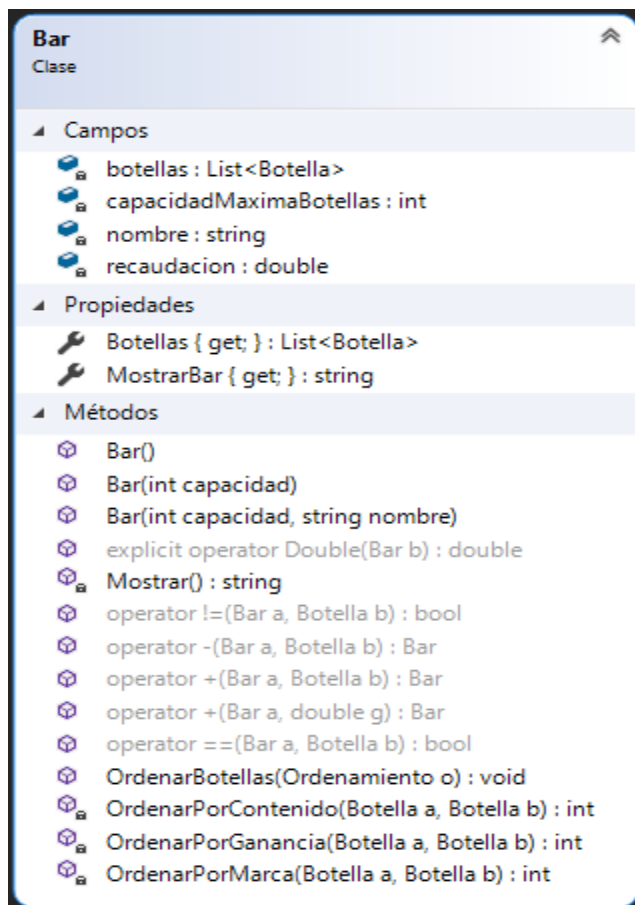
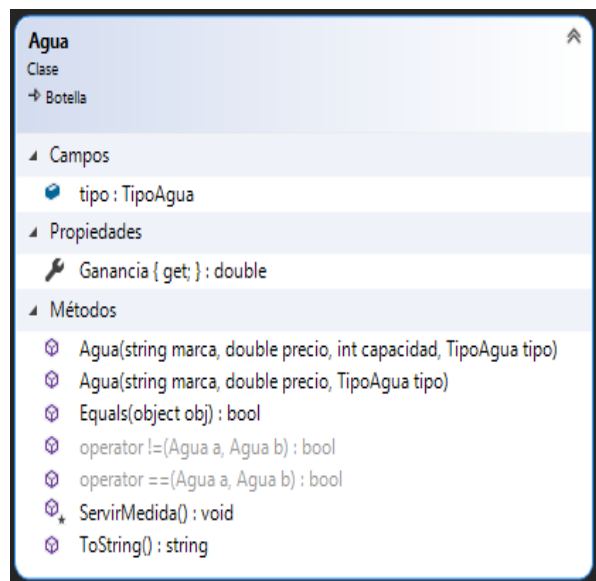
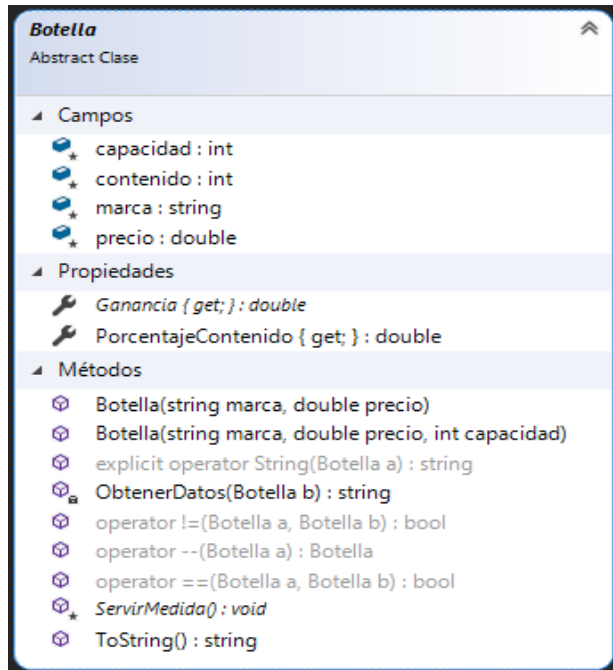


En todos los casos que sean posibles, reutilizar código.

Se tendrá en cuenta las convenciones aplicadas al proyecto. Las entidades estarán en un proyecto de tipo biblioteca de clases.

DIAGRAMA DE CLASES:



Clase Botella (abstracta): *Namespace: Entidades.Botellas*

Tanto la *capacidad* como el *contenido* están expresados en centímetros cúbicos (cc).

Constructor (+ 1 sobrecarga):

Las botellas tendrán una capacidad de 1000 cc de forma predeterminada (al inicializar, el contenido debe ser igual a la capacidad). Reutilizar código.

Propiedades (sólo lectura):

PorcentajeContenido retornará el porcentaje actual del contenido de la botella.

La propiedad abstracta **Ganancia**, retornará la ganancia de la botella.

Métodos:

ServirMedida será abstracto y descuenta del contenido de la botella una unidad de medida.

El método de **clase** *ObtenerDatos* será privado y retornará todos los datos de la botella.

Aplicar polimorfismo en *ToString*, que retornará todos los datos de la botella. Reutilizar código.

Sobrecarga de operadores:

- Igualdad. Retornará *true*, si las marcas y las capacidades son iguales, *false*, caso contrario.
- Explícito. Retornará la marca de la botella que recibe como parámetro.
- Decremento. Saca de la botella que recibe como parámetro una unidad de medida. Reutilizar código.

Clase Agua (deriva de Botella): *Namespace: Entidades.Botellas*

Constructor (+ 1 sobrecarga):

Las botellas de agua tendrán, de forma predeterminada, una capacidad de 500 cc. Reutilizar código.

Propiedades (sólo lectura):

Ganancia, retornará la ganancia de la botella de agua, que será un 25% más del precio total de la botella.

Métodos:

ServirMedida gastará todo el contenido de la botella de agua. La medida de esta botella será la totalidad de la misma.

Aplicar polimorfismo en:

ToString, que retornará todos los datos de la botella de agua. Reutilizar código.

Equals, que retornará *true* si el parámetro recibido es igual a la instancia actual. Reutilizar código.

Sobrecarga de operadores:

- Igualdad. Retornará *true*, si las botellas y los tipos de agua son iguales, *false*, caso contrario.

Clase Cerveza (deriva de Botella): *Namespace: Entidades.Botellas*

La *medida* de cerveza se expresa en centímetros cúbicos (cc).

Constructor (+ 1 sobrecarga):

La medida predeterminada será de un tercio (1/3) de la capacidad de la botella de cerveza. Reutilizar código.

Propiedades (sólo lectura):

Ganancia, retornará la ganancia de cada medida de la botella de cerveza, que será el 50% del precio total de la botella.

Métodos:

ServirMedida descontará, del contenido de la botella de cerveza, una medida. Si una vez descontada, el contenido restante de la botella de cerveza es menor al valor del atributo *medida*, se descartará, vaciando el contenido de la botella.

Aplicar polimorfismo en:

ToString, que retornará todos los datos de la botella de cerveza. Reutilizar código.

Equals, que retornará *true* si el parámetro recibido es igual a la instancia actual. Reutilizar código.

Sobrecarga de operadores:

- Igualdad. Retornará *true*, si las botellas y los tipos de cerveza son iguales, *false*, caso contrario.

Clase Bar: *Namespace: Entidades.Establecimiento*

Constructor (+ 2 sobrecargas):

La cantidad máxima de botellas será de 5 y el valor del nombre será '*Sin nombre*' de forma predeterminada. Reutilizar código.

Propiedades (sólo lectura):

MostrarBar, retornará la información completa del bar. Reutilizar código.

Métodos:

El método de instancia y privado *Mostrar*, retornará una cadena con toda la información del bar, incluyendo el detalle de cada una de sus botellas. Reutilizar código.

El método de instancia *OrdenarBotellas*, ordenará la lista de botellas, de acuerdo al enumerado **Ordenamiento** que recibe como parámetro.

Los métodos privados *OrdenarPorContenido* (de clase), *OrdenarPorGanancia* (de instancia) y *OrdenarPorMarca* (de clase) están asociados al método *OrdenarBotellas*. Reutilizar código.

Sobrecarga de operadores:

- Explícito, retornará la recaudación del bar recibido como parámetro.
- Igualdad, retornará *true*, si es que la botella de agua o la botella de cerveza ya se encuentra en el bar, *false*, caso contrario. Reutilizar código.
- Adición, si el bar (primer parámetro) posee capacidad de almacenar al menos una botella (segundo parámetro) más y esa botella de agua o botella de cerveza no se encuentra en el bar, lo agregará a la colección, caso contrario, no lo agregará. Reutilizar código.
- Adición, acumulará en la recaudación del bar recibido como primer parámetro, la ganancia recibida como segundo parámetro.
- Sustracción, si la botella se encuentra en el bar, se consumirá una medida de esa botella de agua o botella de cerveza. Se acumulará la ganancia correspondiente en el bar y si el porcentaje de contenido es cero, se quitará la botella del bar. Reutilizar código.