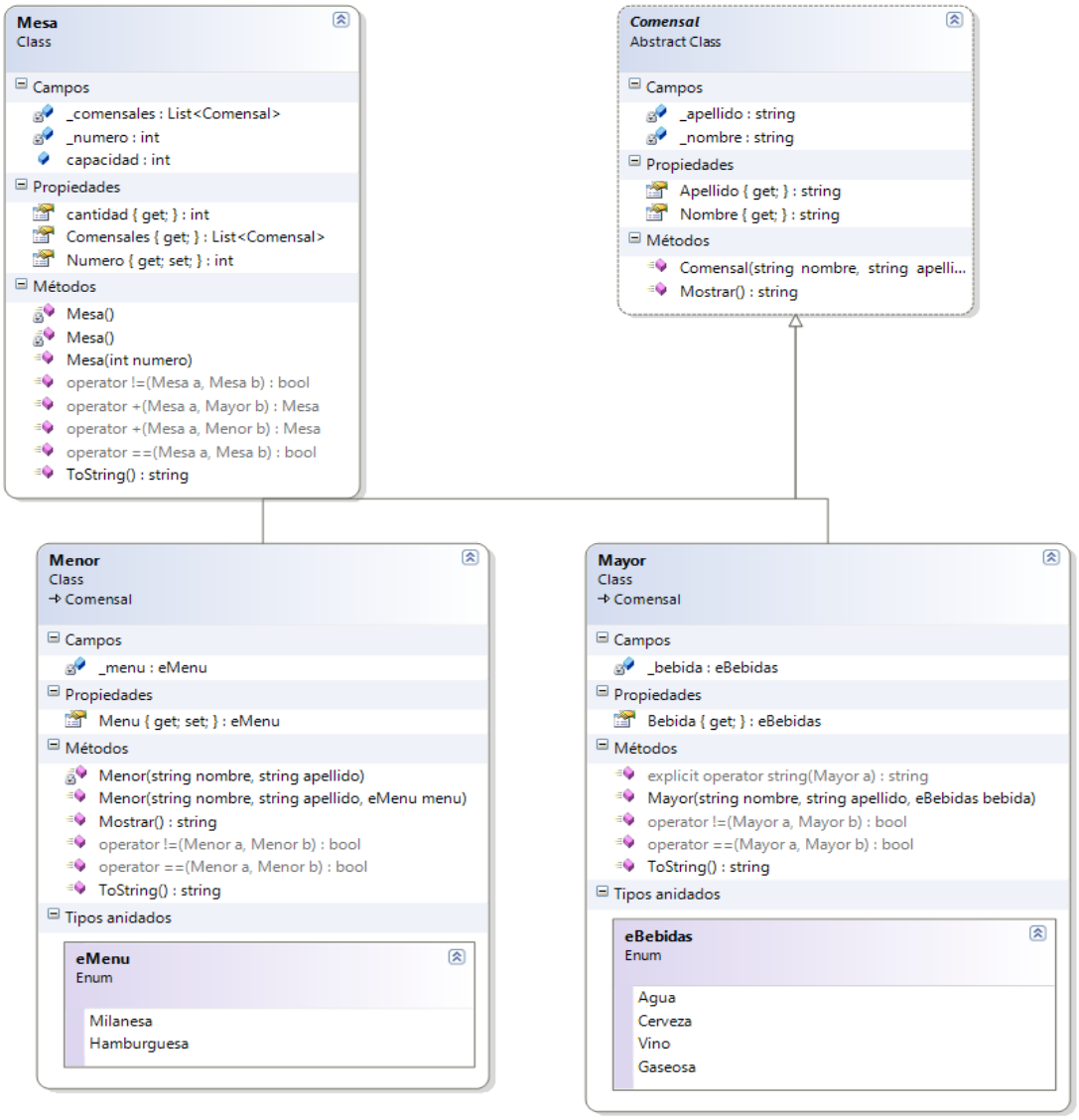
# Universidad Tecnológica Nacional
# Facultad Regional Avellaneda

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos | | | | | | | | | | |
| Materia: Laboratorio II | | | | | | | | | | |
| Apellido: | | | | | Fecha: | | 03/10/2017 | | | |
| Nombre: | | | | | Docente(2): | | Davila - Oggioni | | | |
| División: | | | | | Nota(2): | | | | | |
| Legajo: | | | | | Firma(2): | | | | | |
| Instancia(1): | **PP** | X | **RPP** | | **SP** | | **RSP** | | **FIN** | |

**(1)** Las instancias validas son: 1er Parcial (**PP**), Recuperatorio 1er Parcial (**RPP**), 2do Parcial (**SP**), Recuperatorio 2do Parcial (**RSP**), Final (**FIN**) . Marque con una cruz.

**(2)** Campos a ser completados por el docente.



Los proyectos que no sean identificables, no serán corregidos.

• Sólo se corregirá lo que el alumno entregue de la siguiente forma: o Al finalizar, colocar la carpeta de la Solución completa en un archivo ZIP y dejar este último en el Escritorio de la máquina. Luego presionar el botón de la barra superior, cargar un mensaje y presionar Aceptar. La barra superior deberá cambiar de color.

• En todos los casos que sea posible, reutilizar código.

Se desea administrar las mesas para un evento donde cada una tendrá sus comensales:

*Comensal*: Clase abstracta con dos atributos privados (_nombre y _apellido). El único constructor recibirá dos parámetros. Las propiedades Nombre y Apellido serán de sólo lectura.

Método virtual y public Mostrar().Retornará el nombre y el apellido con el formato "Nombre Apellido". Se deberá utilizar el método Format de la clase String.

*Menor*: Clase pública que hereda de Comensal con un atributos propios (_menu). Contará con dos constructores, uno privado y otro público donde recibe el menú. Implementar el método mostrar(). Retornará toda la información del Menor. Contendrá el enumerado eMenu (Milanesa, Hamburguesa)

Operadores: • Dos Menores serán iguales si comparten nombre, apellido y menu.

Sobreescribir: • Método ToString() para que publique la información del Menor.

• Método Equals para que reutilice ==.

*Mayor*: Clase pública que hereda de Comensal. Operadores: • Dos Mayores serán iguales si comparten nombre y apellido. Contendrá el enumerado eBebidas(Agua, Cerveza, Vino, Gaseosa)

Operadores: • Conversión explicita de Mayor a string, retornando todos sus datos.

Sobreescribir: • Método ToString() para que publique la información del Mayor.

• Método Equals para que reutilice ==.

*Mesa*: Contendrá una lista de Comensales, un numero y atributo de clase capacidad.

Constructores: • De clase que inicializará la capacidad en 12.

• Por defecto privado, será el único lugar donde se inicialice la lista.

• Otro recibirá número.

Propiedades de solo lectura: Número retorna el numero de mesa.

Cantidad que devuelve la cantidad de comensales que hay en la mesa.

Comensales que retorna la lista de comensales.

Sobreescribir: • Método ToString() para que publique la información del Mayor.

Operadores: • Una Mesa será igual si coincide en el número.

• Si un Comensal no forma parte de la lista, se podrá agregar con el +. Siempre que haya lugar en la mesa. El operador + retornara una nueva instancia de mesa con las mismas propiedades y con un comensal mas en la lista

• Conversión implícita a String, debiendo quedar la información con el siguiente formato:

**Mesa: 1 Comensales: 4**

**Juan Perez Hamburguesa Menor**

**Jose Perez Vino**

**Maria Perez Cerveza**

**Vanesa Perez Milanesa Menor**

En el proyecto de formulario, descomentar el código para utilizarlo y completar el método cargarComensales() para que muestre los comensales de la mesa seleccionada