Universidad Tecnológica Nacional
Facultad Regional Avellaneda

Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos										
Materia: Laboratorio de Programación II										
Apellido:					Fecha:		08-10-2020			
Nombre:					Docente(2):		F. Dávila			
División:	2°C				Nota(2):					
Legajo:					Firma(2):					
Instancia(1):	**PP**	X	**RPP**		**SP**		**RSP**		**FIN**	

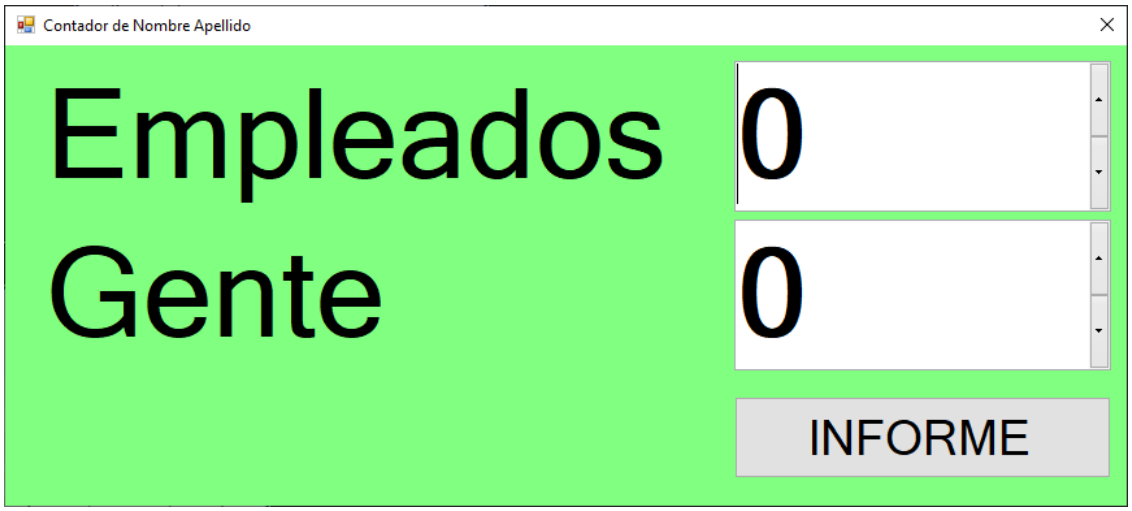
(1) Las instancias validas son: 1er Parcial (**PP**), Recuperatorio 1er Parcial (**RPP**), 2do Parcial (**SP**), Recuperatorio 2do Parcial (**RSP**), Final (**FIN**). Marque con una cruz.
(2) Campos a ser completados por el docente.

IMPORTANTE:

- **2 (dos) errores en el mismo tema anulan su puntaje.**
- La correcta documentación y reglas de estilo de la cátedra serán evaluadas.
- Colocar sus datos personales en el nombre de la carpeta principal y la solución: Apellido.Nombre.Div. Ej: Pérez.Juan.2D. No sé corregirán proyectos que no sea identificable su autor.
- No se corregirán exámenes que no compilen.
- **Reutilizar** tanto código como crean necesario.
- Colocar nombre de la clase (en estáticos), **this** o **base** en todos los casos que corresponda.
- Aplicar los principios de los 4 pilares de la POO.

Se realizará una aplicación que contará la cantidad de persona dentro de un Bar, calculando si permite entrar más gente según la cantidad de empleados que se encuentren disponibles para atenderlos.

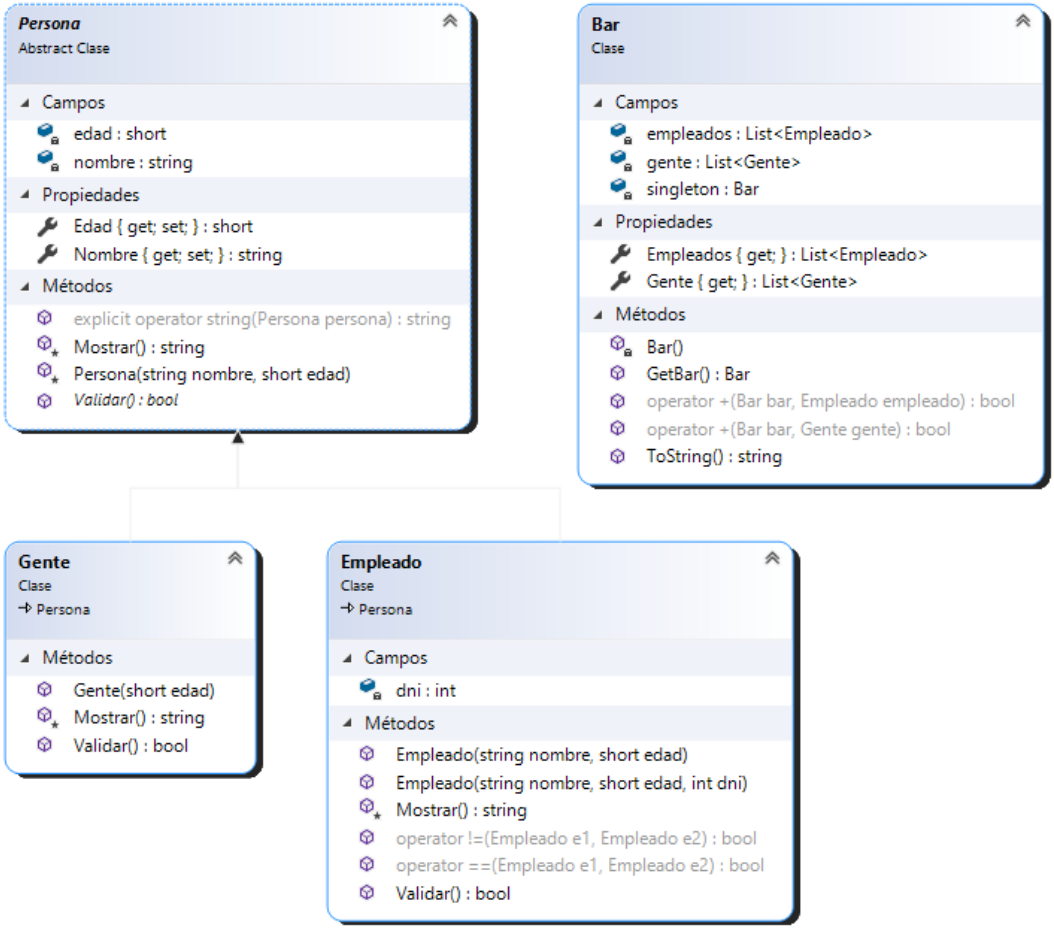
1. ***CuentaGanadoForm***:
    a. El formulario debe iniciar centrado en la pantalla con el siguiente formato:



    b. Quitar los controles de Min y Max del formulario.
    c. Colocar el BackColor en un tono de verde agua.

d. El nombre del formulario debe ser "Contador de " seguido del nombre del alumno.
e. Agregar un atributo del tipo Bar al formulario.
f. Al presionar el botón *INFORME* se abrir un nuevo formulario:
    i. Contará con un RichTextBox llamado *rtbSalidaDeTest*.
    ii. En el mismo se imprimirán los datos del Bar del formulario anterior.
    iii. Este formulario no puede tener código dentro, sólo el método constructor.
    iv. Este formulario deberá permanecer siempre sobre el formulario anterior, hasta que sea cerrado.
g. Los controles tienen que tener nombres según las reglas de estilo vistas.
h. Utilizar el evento ValueChange de los NumericUpDown (la lógica continua en el punto 7).

2. Crear un proyecto del tipo Biblioteca de Clases y colocar el siguiente esquema de clases. Este diagrama puede estar incompleto, faltando algún método que luego se describa:



3. ***Persona***:
    a. Clase abstracta descripta en el diagrama.
    b. *Mostrar* no deberá imprimir información sin cargar (ver lógica de ***Empleado*** y ***Gente***).
4. ***Empleado***:
    a. Herencia de persona, descripta en el diagrama.
    b. El empleado deberá tener nombre y edad de forma obligatoria.
    c. *Validar* chequeará que esta edad sea mayor a 21 años y que su nombre tenga al menos 2 caracteres.
    d. Si un empleado no recibiera el atributo DNI este deberá cargarse con -1. Estos documentos no se deberán mostrar por pantalla en los informes.

e. Dos empleados serán iguales si tienen el mismo nombre y edad.
f. El método *Mostrar* deberá indicar que es un EMPLEADO y luego imprimir su información.

5. ***Gente***:
   a. Herencia de persona, descripta en el diagrama.
   b. La gente podrá no tener nombre, pero debe tener una edad cargada obligatoriamente.
   c. *Validar* chequeará que esta edad sea mayor a 18 años.
   d. El método *Mostrar* deberá indicar que es GENTE y luego imprimir su información.
6. ***Bar***:
   a. Contendrá una lista de Gente y otra de Empleados.
   b. Por cada *Empleado* se aceptarán 10 personas en el *Gente*. Validarlo en el operador +.
   c. +: si están dadas las condiciones, agregará según corresponda.
      i. Para Empleado, no agregar repetidos según el criterio de comparación antes dado.
   d. Sobreescritura ToString: retornará un *String* con toda la información de los *empleados* y *gente* del bar.
7. *ValueChange*:
   a. Cada NumericUpDown contará con su propio manejador del evento.
   b. Al incrementarse el valor se intentará agregar un nuevo elemento a la lista que corresponda.
   c. Al decrementar, eliminar el primer elemento de la lista de Gente o el último elemento de la lista de Empleados.
   d. En principio, agregar siempre el mismo empleado y la misma gente. Utilizar random para la edad.
   e. Punto extra: realizar un formulario para cargar los datos del Empleado y la Gente. El mismo para ambos casos.