

Universidad Tecnológica Nacional

Facultad Regional Avellaneda



Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos

Materia: Laboratorio de Programación II

Apellido:		Fecha:	07-05-2020
Nombre:		Docente ⁽²⁾ :	Dávila-Cerizza-Rodríguez
División:	2°C	Nota ⁽²⁾ :	
Legajo:		Firma ⁽²⁾ :	
Instancia ⁽¹⁾ :	PP	X	RPP
		SP	
		RSP	
		FIN	

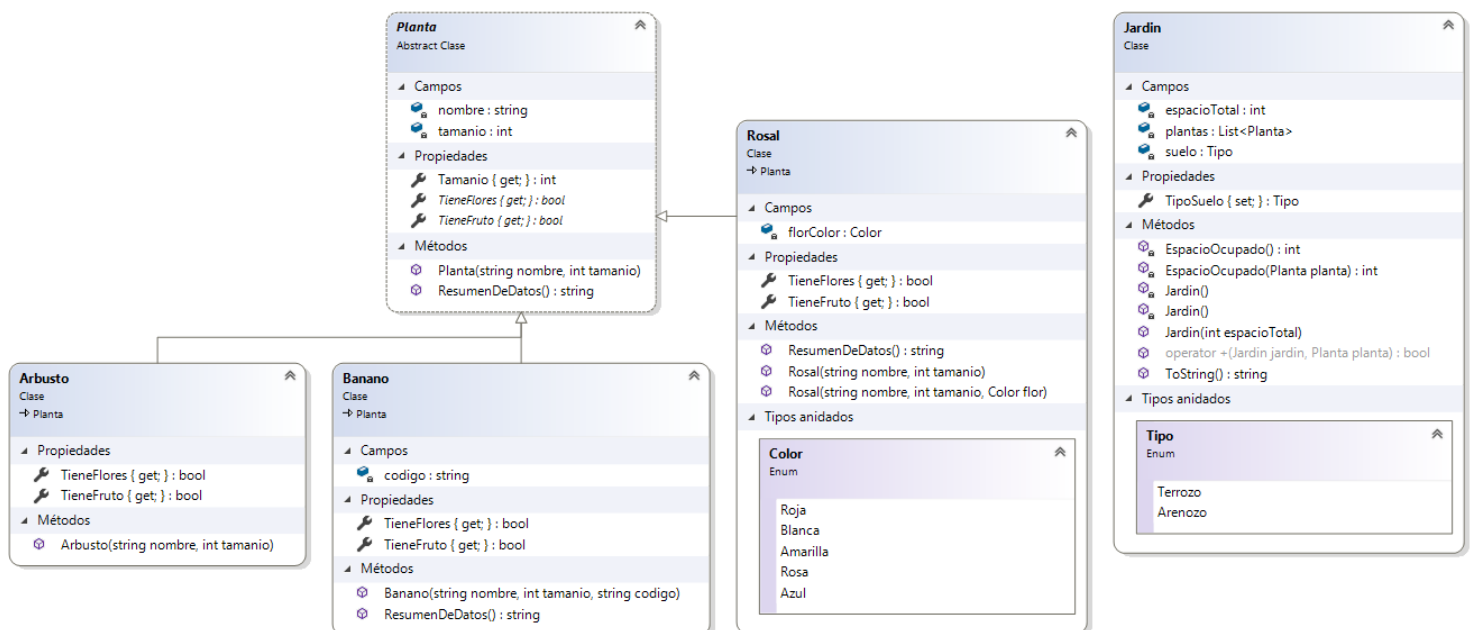
(1) Las instancias validas son: 1^{er} Parcial (PP), Recuperatorio 1^{er} Parcial (RPP), 2^{do} Parcial (SP), Recuperatorio 2^{do} Parcial (RSP), Final (FIN). Marque con una cruz.

(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.

1. Crear un proyecto del tipo Biblioteca de Clases y colocar el siguiente esquema de clases:



2. Clase **Planta**:

- a. Los métodos y propiedades marcados en cursiva son abstractos.
- b. ResumenDeDatos retornará los datos de la Planta utilizando StringBuilder y String.Format.

Completar las palabras en mayúscula con los datos de cada planta según corresponda:

NOMBRE tiene un tamaño de *TAMANIO*

Tiene flores SI/NO

Tiene fruto SI/NO

3. Clase **Rosal**:

- a. Su ResumenDeDatos incorporará el concepto "Flores de color: XXXXX".
- b. Tiene flores y no tiene fruto.

4. Clase **Banano**:

- a. Su ResumenDeDatos incorporará el concepto "Código internacional: XXXXX".
- b. Tiene flores y tiene fruto.

5. Clase **Arbusto**:

- a. No tiene flores ni fruto.

6. Clase **Jardín**:

- a. El Tipo de suelo será estático.
- b. Suelo se inicializará en el constructor estático con el valor Terrozo.
- c. La lista se inicializará en el constructor privado.
- d. El operador + agregará siempre y cuando *espacioTotal* sea suficiente para agregar una nueva planta (espacio ocupado + tamaño de la planta).
- e. EspacioOcupado retornará el espacio total ocupado por todas las plantas del Jardín.
- f. EspacioOcupado(Planta) retornará el total ocupado + el tamaño de la planta recibida como parámetro.
- g. Sobrecribir el método ToString para que retorne los datos del Jardín:

Composición del Jardín: TERROZO/ARENOZO

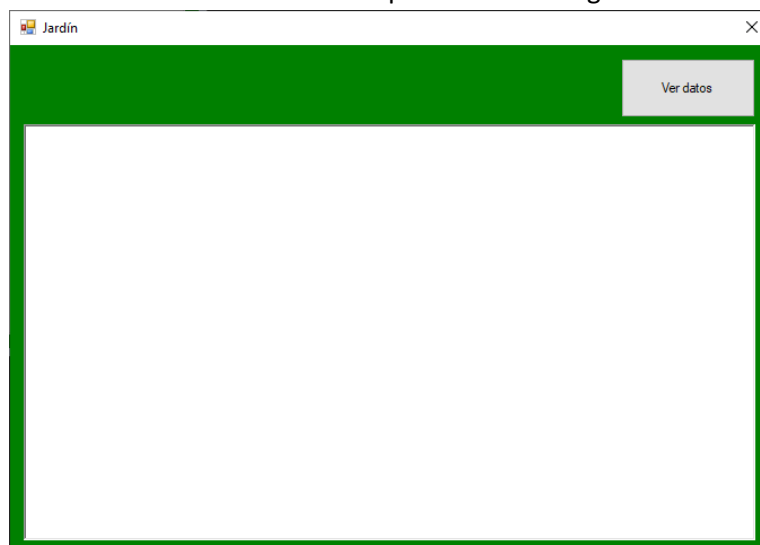
Espacio ocupado XXX de XXX

LISTA DE PLANTAS

NOTA: Aprovechar el concepto de Polimorfismo.

7. Clase **FrmPrincipal**:

- 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 oscuro.
- d. El nombre del formulario debe ser Jardín.
- e. Agregar un atributo del tipo Jardín al formulario.
- f. Agregar el siguiente código al evento *Load* del formulario. Reemplazar el Console.WriteLine por un Message Box con ícono de error y sólo el botón OK:

```
this.jardin = new Jardin(100);
bool pudo = this.jardin + new Arbusto("Arbusto 1", 10);
pudo = this.jardin + new Arbusto("Arbusto 2", 15);
pudo = this.jardin + new Rosal("Rosa 1", 20, Rosal.Color.Amarilla);
pudo = this.jardin + new Rosal("Rosa clásica", 25);
pudo = this.jardin + new Banano("Banano ecuador", 30, "ECU001");
if(!(this.jardin + new Banano("No carga", 1, "ARG028")))
{
    Console.WriteLine("ERROR!");
}
```

- g. Al presionar el botón *Ver datos* se mostrará en el RichTextBox *rtbSalidaDeTest* los datos del jardín.