

Universidad Tecnológica Nacional
Facultad Regional Avellaneda



Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos

Materia: Laboratorio de Programación II

Apellido:		Fecha:	2021-05-13
Nombre:		Docente ⁽²⁾ :	Dávila
División:	2°C	Nota ⁽²⁾ :	
Legajo:		Firma ⁽²⁾ :	
Instancia ⁽¹⁾ :	<div style="display: flex; justify-content: space-around;"> <div>PP</div> <div>X</div> <div>RPP</div> <div></div> <div>SP</div> <div></div> <div>RSP</div> <div></div> <div>FIN</div> <div></div> </div>		

(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:

Personaje

Abstract Clase

Campos

- listaHabilidades : List<EHabilidades>
- nombre : string

Propiedades

- ListaHabilidades { get; } : string
- Nombre { get; } : string

Métodos

- operator !=(List<Personaje> listaPersonajes, Personaje personaje) : bool
- operator +(List<Personaje> listaPersonajes, Personaje personaje) : List<P...
- operator ==(List<Personaje> listaPersonajes, Personaje personaje) : bool
- Personaje()
- Personaje(string nombre, List<EHabilidades> habilidades)
- ToString() : string

Avenger

Clase

→ Personaje

Campos

- equipamiento : EEquipamiento

Propiedades

- Nombre { get; } : string

Métodos

- Avenger(string nombre, List<EHabilidades> hab, EEquipamiento equipo)
- ToString() : string

Enemigo

Clase

→ Personaje

Campos

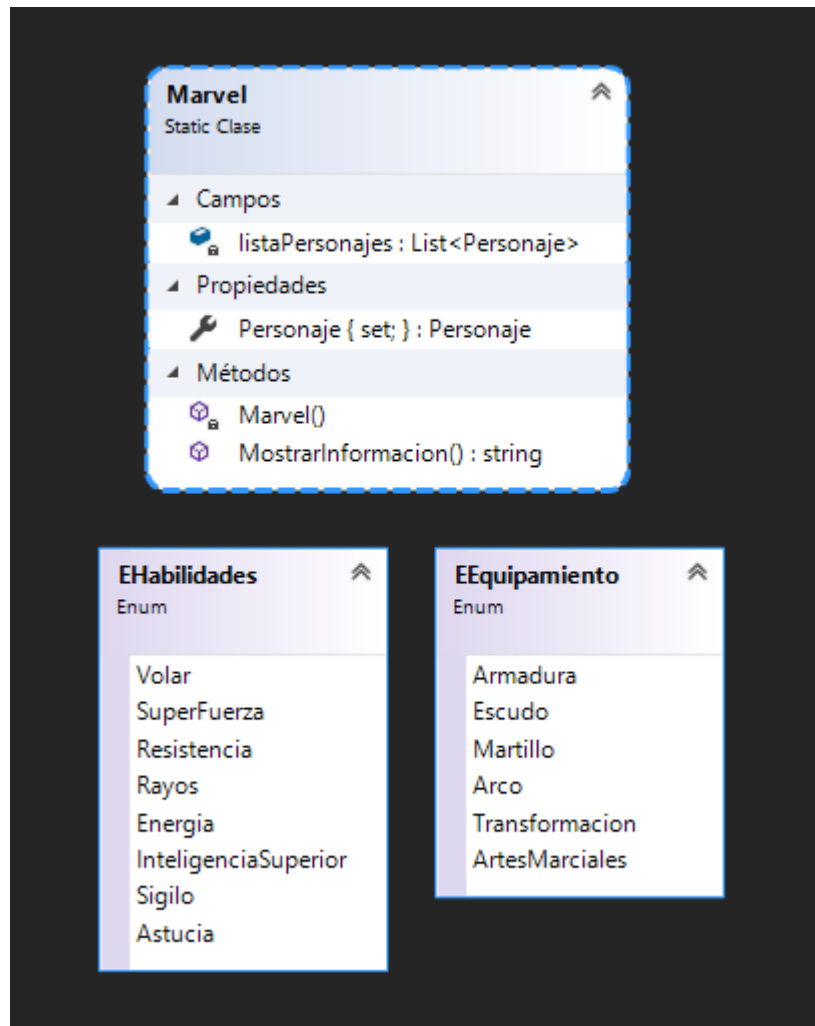
- objetivo : string

Propiedades

- Nombre { get; } : string

Métodos

- Enemigo(string nombre, List<EHabilidades> hab, string objetivo)
- ToString() : string



2. Clase **abstract Personaje**:

- a. La lista de habilidades se Instanciará en el constructor privado.
- b. La propiedad privada `ListaHabilidades` retornará un string con todas las habilidades de ese personaje. Utilizar `StringBuilder`.
- c. La **sobrecarga del operador ==** comparará una lista de personajes con un personaje para determinar si ese personaje se encuentra o no en la lista. Este operador comparará por nombre y por tipo de objeto. Utilizar `GetType()`.
- d. La **sobrecarga operador +** agregará un personaje a la lista de personajes que se pasa por parámetro, siempre y cuando no exista previamente ese personaje en la lista. Retornará la lista.
- e. Sobrecarga del `ToString`: Devolverá toda la información del personaje, tal como:
 - i. Nombre
 - ii. `ListaHabilidades`.Utilizar `StringBuilder`.

3. Clase Avenger

- a. La propiedad Nombre debe ser sobrescrita para que retorne
"Mi nombre es XXXX y si no puedo y si no puedo proteger la tierra, la vengaré"
Reemplazar XXXX con el nombre del personaje.
- b. Sobrecarga ToString devolverá todos los datos de ese Avenger. Debe incluir los datos de la clase base y el equipamiento.
Utilizar StringBuilder.

4. Clase Enemigo

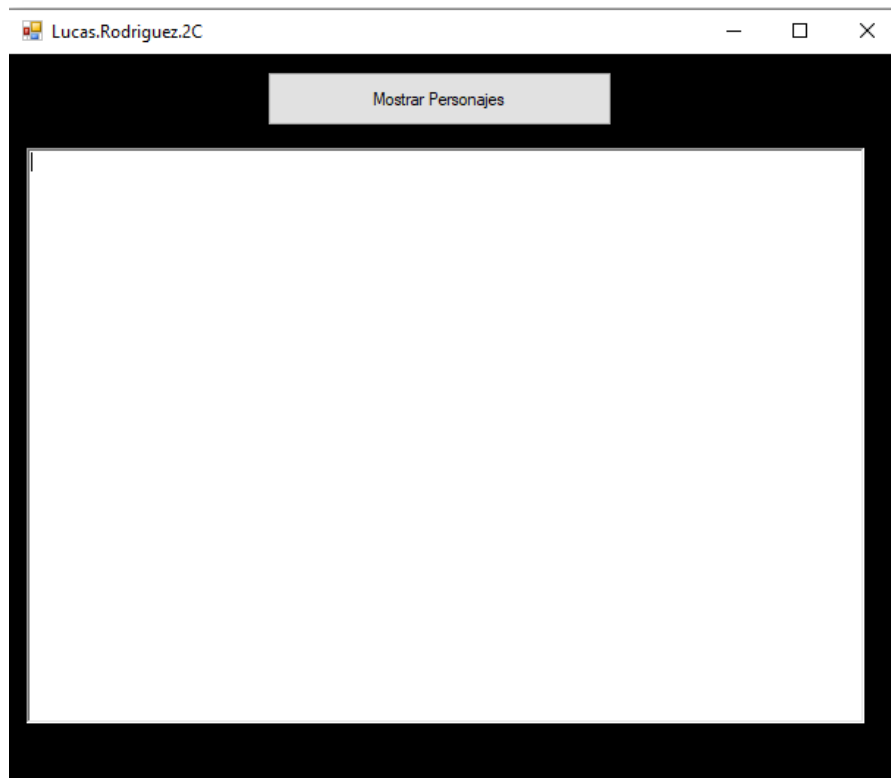
- a. La propiedad Nombre debe ser sobrescrita para que retorne
"Soy XXXX y los voy a hacer puré"
Reemplazar XXXX con el nombre del personaje.
- b. Sobrecarga ToString devolverá todos los datos de ese Avenger. Debe incluir los datos de la clase base y el objetivo. Utilizar StringBuilder.

5. Clase Estática Marvel

- a. La lista privada llamada listaPersonajes será instanciada en el constructor de la clase.
- b. La propiedad Personaje agregará un personaje siempre y cuando ese personaje **NO** exista en la lista. Reutilizar código previamente desarrollado.
- c. Método MostrarInformación devolverá un string con la información COMPLETA de todos los personajes de la lista. Si dicho personaje es un AVENGER mostrará un encabezado al estilo
***** AVENGER *****
Si dicho personaje es un Enemigo, mostrará el encabezado al estilo
***** ENEMIGO *****
Utilizar StringBuilder.

6. Clase **FrmPrincipal**

- a. El formulario debe tener el siguiente formato:



- b. Colocar el BackColor en un tono negro.
- c. El nombre del formulario debe ser **FormularioPrincipal**.
- d. El título del formulario debe ser **NOMBRE.APELLIDO.DIVISION**
- e. Agregar el siguiente código al evento *Load* del formulario.

```
Marvel.Personaje = new Avenger("Anthony Stark", EOrigen.Terricola, new List<EHabilidades>() {  
EHabilidades.InteligenciaSuperior }, EEquipamiento.Armadura);  
Marvel.Personaje = new Avenger("Anthony Stark", EOrigen.Terricola, new List<EHabilidades>() {  
EHabilidades.InteligenciaSuperior }, EEquipamiento.Armadura);  
Marvel.Personaje = new Avenger("Dr Banner", EOrigen.Terricola, new List<EHabilidades>() {  
EHabilidades.InteligenciaSuperior }, EEquipamiento.Transformacion);  
Marvel.Personaje = new Avenger("Dr Banner", EOrigen.Terricola, new List<EHabilidades>() {  
EHabilidades.InteligenciaSuperior }, EEquipamiento.Transformacion);  
Marvel.Personaje = new Avenger("Natasha Romanoff ", EOrigen.Terricola, new  
List<EHabilidades>() { EHabilidades.Sigilo, EHabilidades.Astucia }, EEquipamiento.ArtesMarciales);  
Marvel.Personaje = new Avenger("Thor", EOrigen.Asgardiano, new List<EHabilidades>() {  
EHabilidades.Rayos, EHabilidades.Volar }, EEquipamiento.Martillo);
```

```
Marvel.Personaje = new Enemigo("Thanos", EOrigen.Extraterreste, new List<EHabilidades>() {  
EHabilidades.SuperFuerza, EHabilidades.Astucia, EHabilidades.Resistencia }, "Obtener las infinity  
stones y un te de vainilla");
```

```
Marvel.Personaje = new Enemigo("Ultron", EOrigen.Maquina, new List<EHabilidades>() {  
EHabilidades.SuperFuerza, EHabilidades.Astucia, EHabilidades.Volar }, "Exterminar a los humanos");
```

```
Marvel.Personaje = new Enemigo("Loki", EOrigen.Asgardiano, new List<EHabilidades>() {  
EHabilidades.Astucia }, "Dominar los 9 reinos");
```

- f. Al presionar el botón **MostrarPersonajes** se mostrará en el RichTextBox **rtbPersonajes** los datos de todos los personajes.

