

Universidad Tecnológica Nacional Facultad Regional Avellaneda									
Materia: Laboratorio III									
Apellido:					Fecha:				
Nombre:					Docente:				
División:					Nota:				
Legajo:					Firma:				
Instancia:	PP	X	RPP	SP		RSP		FIN	

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

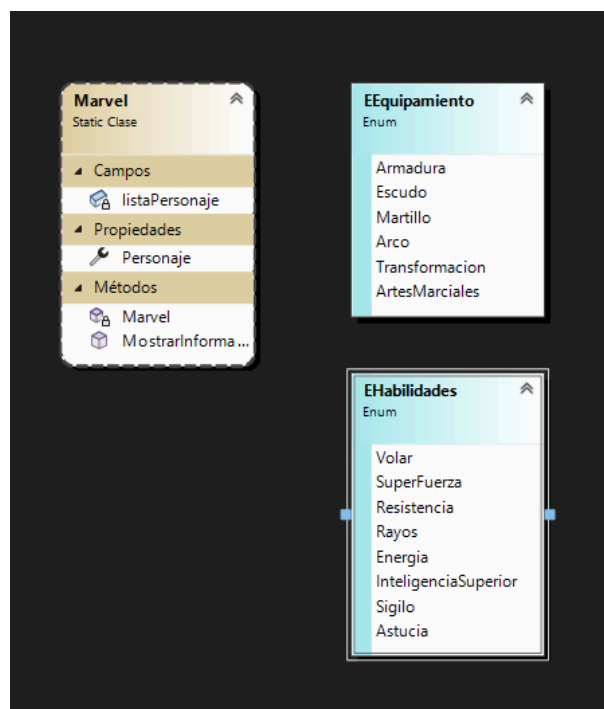
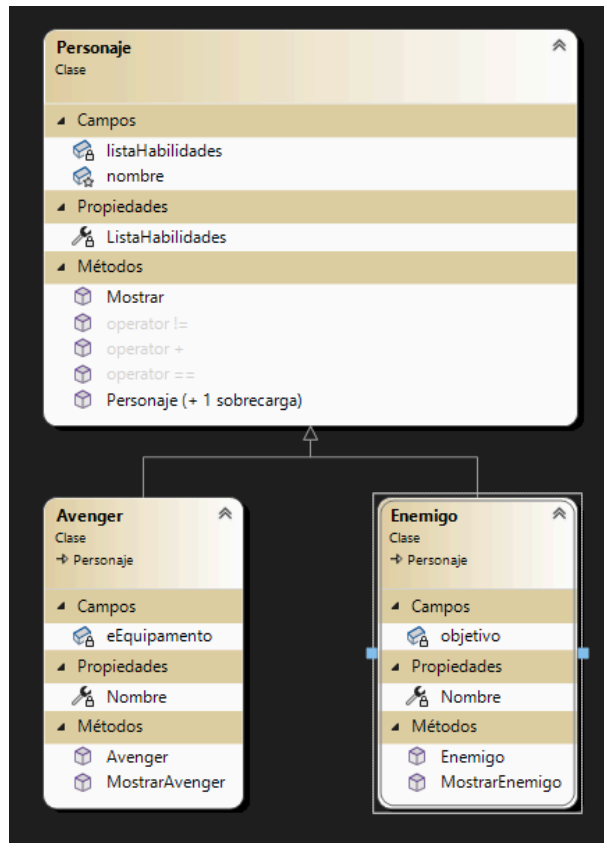
IMPORTANTE:

- 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 se 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 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. Mostrar: Devolverá toda la información del personaje, tal como:
 - i. Nombre
 - ii. ListaHabilidades. Utilizar StringBuilder.

3. Clase Avenger

- a. La propiedad Nombre retorna
"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. MostrarAvenger devolverá todos los datos de ese Avenger. Debe incluir los datos de la clase base y el equipamiento. Utilizar Stringbuilder.

4. Clase Enemigo

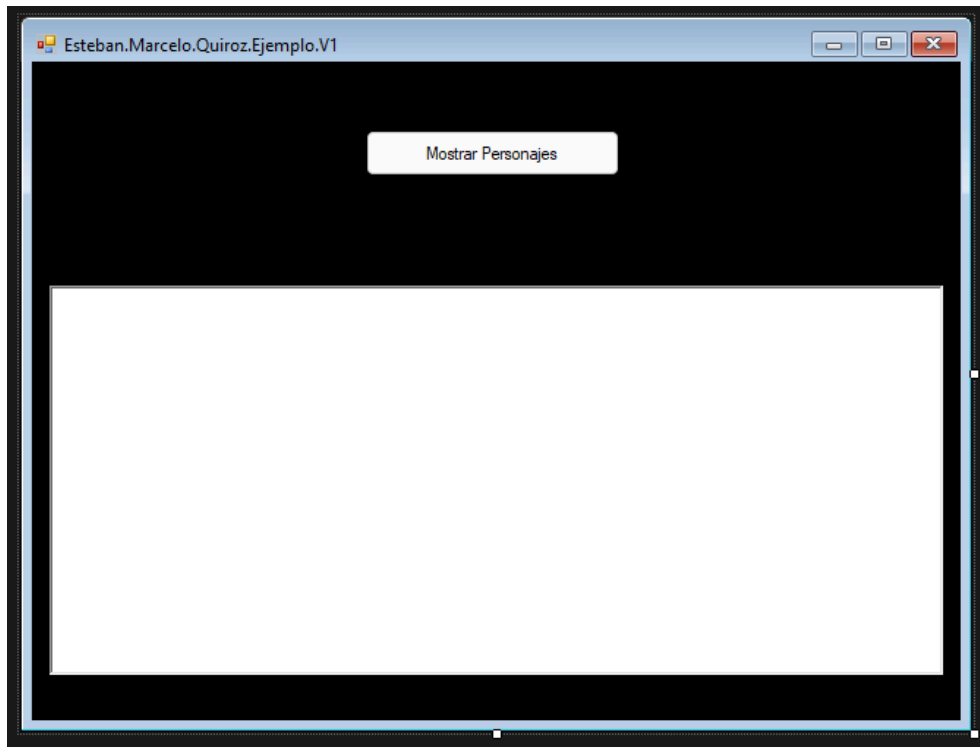
- a. La propiedad Nombre retorna
"Soy XXXX y los voy a hacer puré"
Reemplazar XXXX con el nombre del personaje.
- b. MostrarEnemigo devolverá todos los datos de ese Enemigo. Debe incluir los datos de la clase base y el objetivo. Utilizar Stringbuilder.

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. MétodoMostrarInformación 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

El formulario debe tener el siguiente formato:



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

```
// AVENGERS
```

```
Marvel.Personaje = new Avenger("AnthonyStark", new List<EHabilidades>() {  
    EHabilidades.InteligenciaSuperior }, EEquipamiento.Armadura);
```

```
Marvel.Personaje = new Avenger("AnthonyStark", new List<EHabilidades>() {  
    EHabilidades.InteligenciaSuperior }, EEquipamiento.Armadura);
```

```
Marvel.Personaje = new Avenger("Dr Banner", new List<EHabilidades>() {  
    EHabilidades.InteligenciaSuperior }, EEquipamiento.Transformacion);
```

```
Marvel.Personaje = new Avenger("Dr Banner", new List<EHabilidades>() {  
    EHabilidades.InteligenciaSuperior }, EEquipamiento.Transformacion);
```

```
Marvel.Personaje = new Avenger("NatashaRomanoff ", new List<EHabilidades>() {  
    EHabilidades.Sigilo, EHabilidades.Astucia }, EEquipamiento.ArtesMarciales);
```

```
Marvel.Personaje = new Avenger("Thor", new List<EHabilidades>() {  
    EHabilidades.Rayos, EHabilidades.Volar }, EEquipamiento.Martillo);
```

```
Marvel.Personaje = new Avenger("Thor", new List<EHabilidades>() {  
    EHabilidades.Rayos, EHabilidades.Volar }, EEquipamiento.Martillo);
```

```
//VILLANOS
```

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

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

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

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

