

Apellido:		Fecha:	12-08-2021
Nombre:		Docente ⁽²⁾ :	
División:		Nota ⁽²⁾ :	
Legajo:		Firma ⁽²⁾ :	
Instancia ⁽¹⁾ :	PP	RPP	SP
		RSP	FIN
			X

(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.

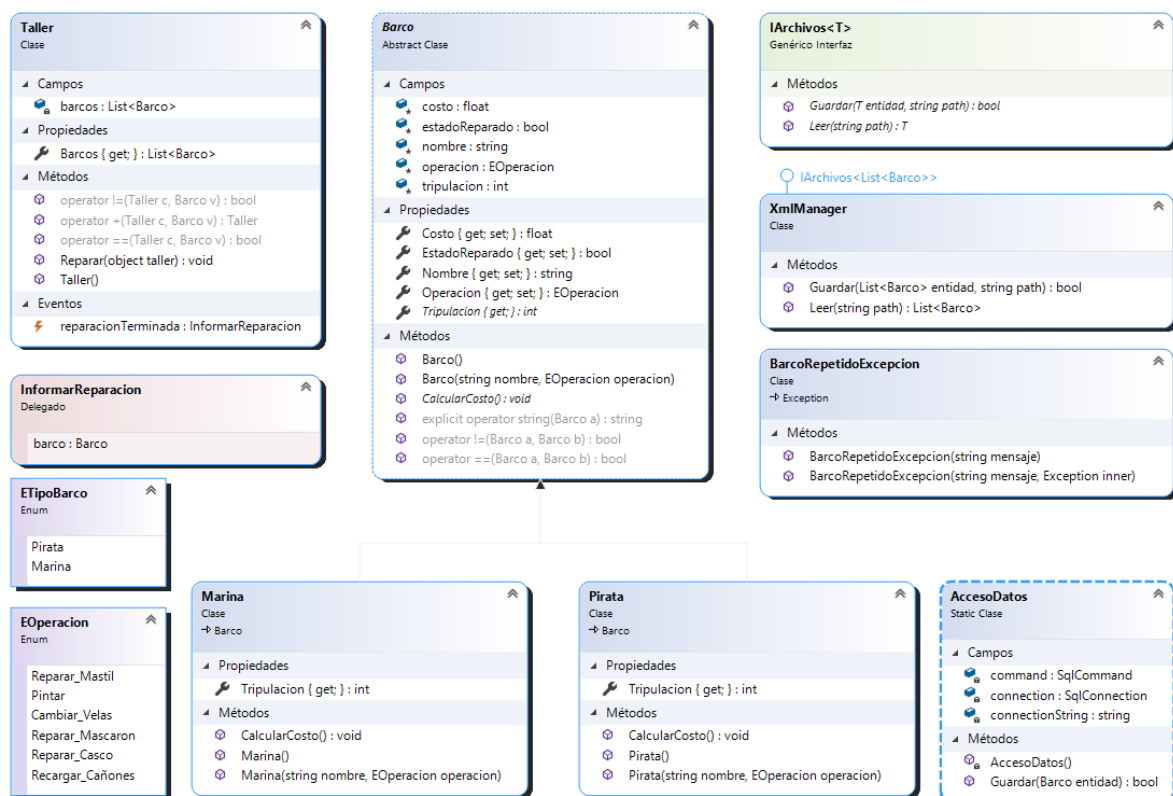
(2) Campos a ser completados por el docente.

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.

Se desea desarrollar una aplicación para un taller de reparación de barcos. Para ello se debe:

1. Crear un proyecto de tipo biblioteca de clases con el siguiente esquema:



2. Clase Barco:
 - a. Clase Abstracta. Los miembros marcados *en cursiva* deberán ser declarados como miembros *abstractos*.
 - b. El constructor sin parámetros no hará nada.
 - c. Dos barcos serán iguales si comparten el mismo nombre.
 - d. La conversión explícita deberá retornar el costo con el signo '\$' por delante.
3. Clase Pirata:
 - a. Sobrescribir la propiedad Tripulacion. Si el barco no tiene tripulación, generar un numero random entre 10 y 30.
 - b. Sobrescribir el método CalcularCosto. Generar un numero random entre 2000 y 12000 y asignar al atributo.
4. Clase Marina:
 - a. Sobrescribir la propiedad Tripulacion. Si el barco no tiene tripulación, generar un numero random entre 30 y 60.
 - b. Sobrescribir el método CalcularCosto. Generar un numero random entre 5000 y 25000 y asignar al atributo.
5. Clase Taller:
 - a. El constructor será el encargado de instanciar la lista.
 - b. La sobrecarga del operador == retornara true si el barco se encuentra en la lista.
 - c. La sobrecarga del operador + validará que el barco no se encuentre en la lista y lo agregará. Caso contrario lanzar una excepción BarcoRepetidoException.
 - d. El método Reparar será ejecutado en un hilo secundario y deberá:
 - i. Validar que el objeto que recibe como parámetro sea de tipo taller.
 - ii. Recorrer la lista de barcos. Si el barco no fue reparado (tener en cuenta la propiedad EstadoReparado):
 1. Calcular el costo de la reparación.
 2. Guardar la reparación en la BD.
 - iii. Informar por medio del evento que la reparación terminó.
 - iv. Realizar un Sleep de 3 segundos por cada reparación.
 - v. Cambiar el estado del barco a Reparado.
6. Interfaz:
 - a. Generar una Interfaz genérica IArchivos.
7. Clase XmlManager:
 - a. Implementar IArchivos. Esta clase será la responsable de serializar y deserializar una lista de Barcos.
8. Clase AccesoDatos:
 - a. Se utilizará para guardar en la BD por cada reparación: El mensaje: **\$"Se reparo el {nombre} a un costo de {costo} berries"**. Y el nombre del alumno.
 - b. Script para crear BD:

```
CREATE DATABASE Final;
GO
USE Final;
```

```
GO
CREATE TABLE Final.dbo.reparaciones(
id INT IDENTITY (1,1) PRIMARY KEY,
mensaje VARCHAR (255) NOT NULL,
alumno VARCHAR (255) NOT NULL,
);
GO
```

9. Test Unitarios:

- a. Probar que se lance la excepción del punto 5.c.
- b. Probar que se guarde el archivo Xml.

10. Formularios:

- a. Cuanta con los comentarios necesarios sobre los métodos donde deberán realizar determinadas acciones. Estos comentarios inician de la siguiente forma **//TODO**.
 - i. FrmPrincipal:
 1. El evento CLICK del botón Cargar Barco agregará el barco al taller si DialogResult retorna OK. Capturar las posibles Excepciones por barcos repetidos.
 2. El evento CLICK del botón Guardar, guardará el taller en un archivo XML en el mismo directorio que el proyecto.
 3. Al intentar cerrar el formulario preguntar si se desea salir.
 - ii. FrmBarco:
 1. El evento CLICK del botón Cargar instanciará el barco y retornará OK.
 - iii. FrmReparacion:
 1. Recibe un Taller como parámetro.
 2. El evento LOAD deberá:
 - a. suscribir el manejador al evento.
 - b. Instanciar el hilo, iniciarlo y pasarle los parámetros necesarios.
 - c. ImprimirTicket será el método manejador, responsable de mostrar la información en el control correspondiente.
 - d. El evento CLOSING será el responsable de cerrar el proceso antes de salir del formulario.