

**Universidad Tecnológica Nacional
Facultad Regional Avellaneda**



Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos										
Materia: Laboratorio de Programación II										
Apellido:					Fecha:	20/11/2018				
Nombre:					Docente ⁽²⁾ :	F.Dávila/D.Boullon				
División:	2ºD				Nota ⁽²⁾ :					
Legajo:					Firma ⁽²⁾ :					
Instancia ⁽¹⁾ :	PP		RPP		SP	X	RSP		FIN	

(1) Las instancias validas son: 1º Parcial (PP), Recuperatorio 1º Parcial (RPP), 2º Parcial (SP), Recuperatorio 2º Parcial (RSP), Final (FIN). Marque con una cruz.

(2) Campos a ser completados por el docente.

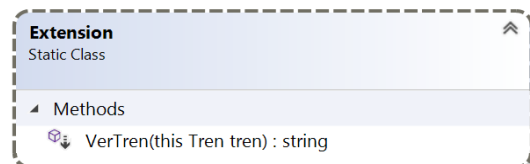
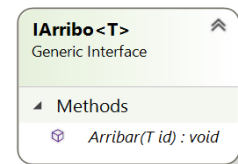
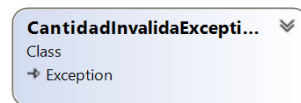
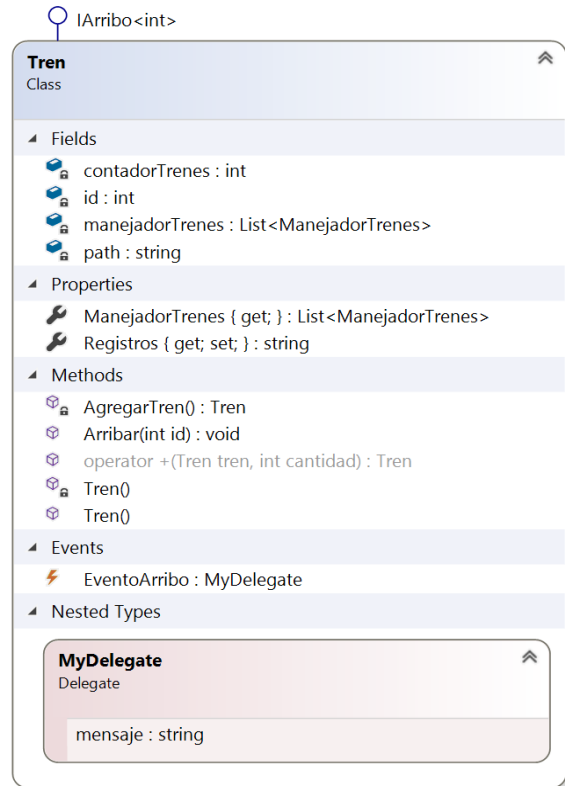
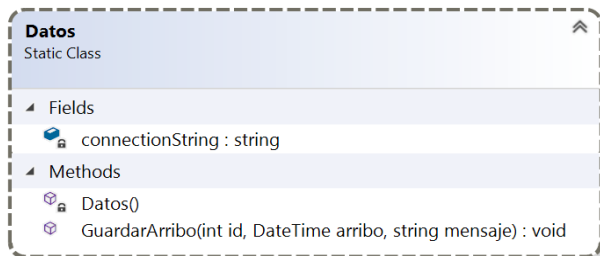
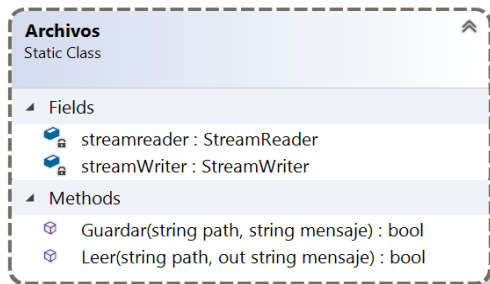
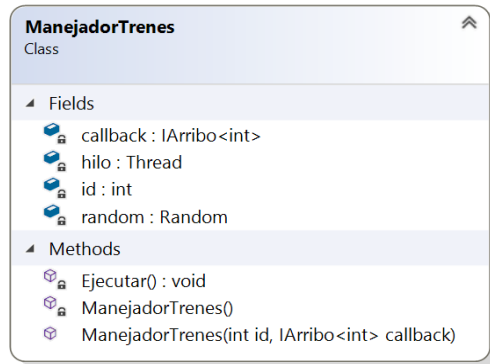
IMPORTANTE:

- **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 del proyecto principal, colocando:
Apellido.Nombre.Departamento. Ej: Pérez.Juan.2D. No se corregirán proyectos que no sea identificable su autor.
- **TODAS** las clases deberán ir en una Biblioteca de Clases llamada Entidades.
- 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.

TIEMPO MÁXIMO PARA RESOLVER EL EXAMEN 90 MINUTOS.

Para generar la base de datos se debe ejecutar la consulta que se encuentra en el archivo "script.sql".

1. En el Proyecto "Clases" agregar las siguientes clases:



Clase ManejadorTrenes

- Constructores:**
 - Dos parámetros:** recibe un **id (int)** y el **callback (IArribo<int>)**.
 - Deberá instanciar el atributo Hilo pasándole el método **Ejecutar**.
 - Iniciará al atributo **Hilo**.
- Atributo:**
 - random (static random).** El mismo se inicializará en un constructor de clase.
 - hilo (Thread)** será el que ejecutará el método **Ejecutar**.
- Métodos:**
 - Ejecutar** frenará el código durante un tiempo aleatorio de entre 1 y 5 segundos. Luego de transcurrir este tiempo, utilizará el método **Arribar** del **callback** pasando como parámetro el atributo id.

Clase Tren

- Debe implementar la interface **IArribo<int>**.

- **Constructores:**
 - **De instancia, sin parámetros,** deberá asignar el valor de `contadorTrenes` al `id`, luego aumentar en uno dicha variable e instanciará la lista de **ManejadorTrenes**.
 - **Estático:** asignará un 0 al atributo estático `contadorTrenes` y `path` con la ruta a un archivo "RegistroDeTrenes.txt" en el escritorio.
- **Properties:**
 - **Registros (string):**
 - Set:** genera o agrega información al archivo descrito en *path* y también guardará en la tabla **trenes** de la **base de datos** usando el método **GuardarArribo**.
 - Get:** retornará el contenido del archivo.
 - **ManejadorTrenes:** retorna el atributo `manejadorTrenes`.
- **Método:**
 - **AgregarTren** hará los siguientes pasos, en el siguiente orden:
 - Generará un nuevo objeto del tipo **Tren**.
 - Le asignará el evento **EventoArribo** de `this` al objeto creado (ej: `tren.EventoArribo = this.EventoArribo`).
 - Crearé un nuevo **ManejadorTrenes** y en el constructor le pasará el `id` del tren y el tren creado.
 - **Arribar (Implementación IArribo)** hará los siguientes pasos, en el siguiente orden:
 - Crearé un mensaje con el siguiente formato: "Llegó el tren {0}."
 - Guardará el mensaje en la property **Registros**.
 - Ejecutará el evento `EventoArribo`.
- El operador **+** lanzará la excepción **CantidadInvalidaException** en el caso de que la cantidad sea menor a 1. Si cantidad es mayor a 0, deberá agregar tantos trenes como indique dicha cantidad.

Clase estática Dato

- **Atributos:**
 - **connectionString (static string),** debe contener la información necesaria para conectarse a la base de datos.
- **Métodos:**
 - **GuardarArribo,** guarda los parámetros que recibe en la tabla **trenes**.
- **Constructor:**
 - **Estático sin parámetros,** le asigna el valor al `connectionString`.

Clase estática Archivos.

- **Atributos:**
 - **streamreader (static StreamReader).**
 - **streamWriter (static StreamWriter).**
- **Métodos:**
 - **Leer:**
 - Recibe el `path` de un archivo, lo lee completo y carga un parámetro de tipo `string` que recibe por referencia (**out**) con el contenido.
 - Retorna un booleano que indica el éxito de la operación.
 - **Guardar:**
 - Recibe el `path` de un archivo, lo lee completo y agrega el mensaje que recibe como parámetro al final del archivo.

- Retorna un booleano que indica el éxito de la operación.

- **Constructor:**

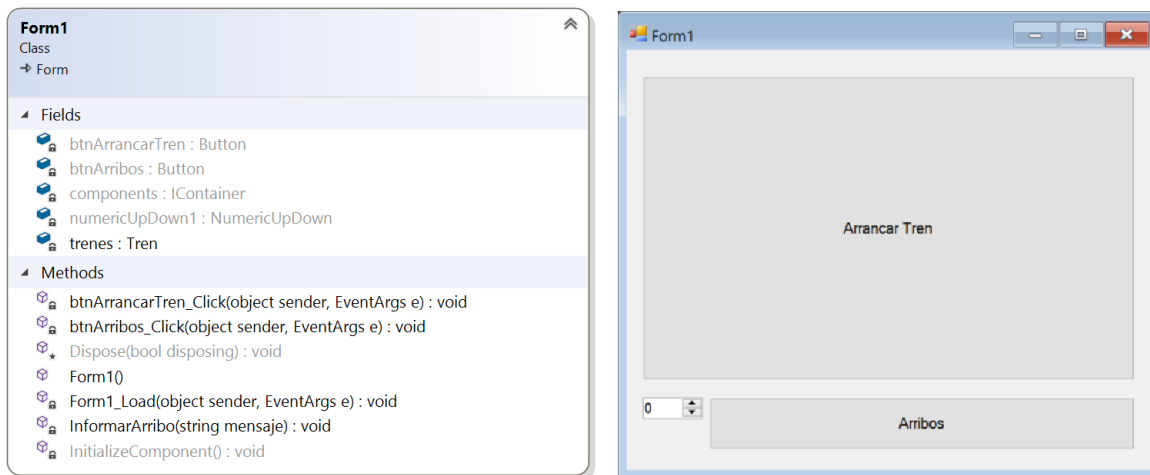
- **Estático sin parámetros**, le asigna el valor al connectionString.

Clase Extension

- **Metodo de extensión:**

- **VerTren:** Método que extiende la clase tren, debe usar la property Registros, de tren para retornar un string.

2. Generar un proyecto con el nombre del alumno del tipo Windows Form. El formulario deberá verse exactamente como este:



Form1

- **Método:**

- **InformarArribo** recibe un string llamado mensaje y lo muestra en un MessageBox
 - **Form1_Load**, en el load del formulario se asignará el método **InformarArribo** al evento **EventoArribo** de trenes.
 - **Botón ArrancarTren:**
 - i. Sumará la cantidad de trenes indicada en el **numericupdown**.
 - ii. Deberá agregar un **nuevo tren** al atributo **trenes** con la **sobrecarga** del +.
 - iii. En caso de error, se mostrará mediante un MessageBox.

- Al presionar el botón **Arribos** se deberá mostrar por pantalla la lista de arribos guardada por **trenes** (mediante el atributo registros).


3. Generar un proyecto de Test Unitario y hacer un método de prueba con el nombre Sobrecargas que pruebe la sobrecarga + Trenes. Deberá cumplirse el siguiente caso:

Caso:

- Tren tren = new Tren();
- tren += -1;

El test debe corroborar que lance una **CantidadInvalidaException**.

Al finalizar, colocar la carpeta de la Solución completa en un archivo ZIP que deberá tener como nombre Apellido.Nombre.division.zip y dejar este último en el Escritorio de la máquina.

Luego presionar el botón  de la barra superior, colocar un mensaje y apretar **Aceptar**. Finalmente retirarse del aula y aguardar por la corrección.