

# Universidad Tecnológica Nacional Facultad Regional Avellaneda



Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos

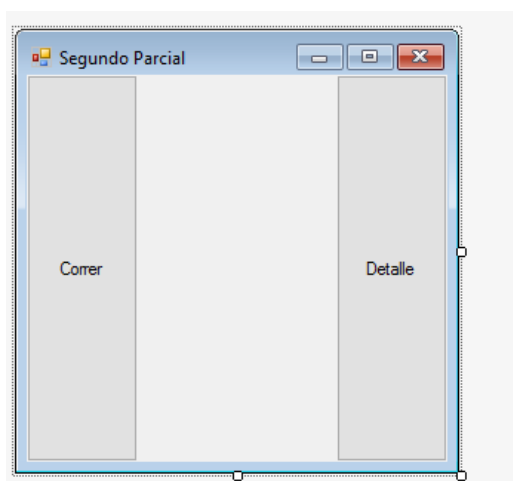
**Materia: Laboratorio II**

Apellido:		Fecha:	21/11/2017
Nombre:		Docente <sup>(2)</sup> :	Davila - Oggioni
División:		Nota <sup>(2)</sup> :	
Legajo:		Firma <sup>(2)</sup> :	
Instancia <sup>(1)</sup> :	<div style="display: flex; justify-content: space-around;"> <span><b>PP</b></span> <span></span> <span><b>RPP</b></span> <span></span> <span><b>SP</b></span> <span>x</span> <span><b>RSP</b></span> <span></span> <span><b>FIN</b></span> <span></span> </div>		

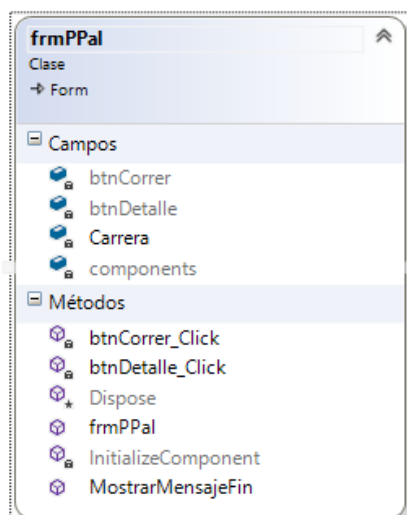
(1) Las instancias validas son: 1<sup>er</sup> Parcial (**PP**), Recuperatorio 1<sup>er</sup> Parcial (**RPP**), 2<sup>do</sup> Parcial (**SP**), Recuperatorio 2<sup>do</sup> Parcial (**RSP**), Final (**FIN**) . Marque con una cruz.

(2) Campos a ser completados por el docente.

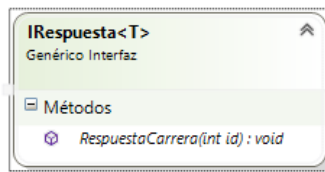
1. Generar un proyecto con el nombre del alumno del tipo Windows Form. El formulario deberá verse exactamente como este:



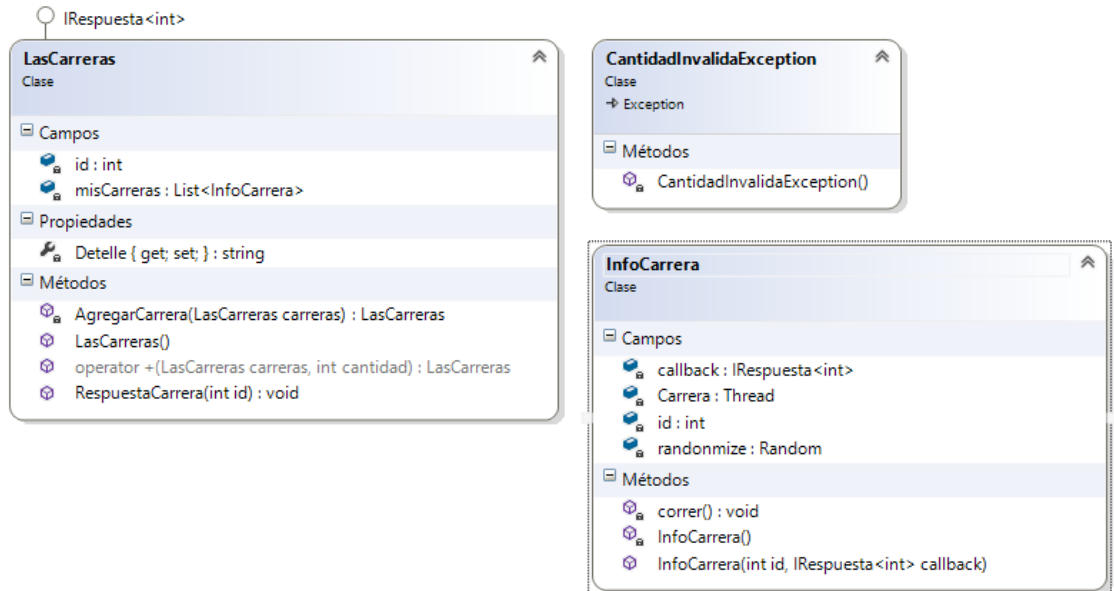
2. Y la clase del formulario tendrá el siguiente formato:



3. Generar un proyecto llamado Interfaces y dentro colocar la siguiente interfaz:



4. Generar un proyecto llamado Entidades y dentro colocar las siguientes clases:



5. Formulario:

- a. Crearemos el atributo carrera.
  - b. El evento AvisoFin de carrera deberá estar relacionado con el método MostrarMensajeFin del formulario.
  - c. MostrarMensajeFin mostrará por pantalla el mensaje recibido.
  - d. Al presionar el botón Comenzar se deberá, mediante la sobrecarga del +, agregar una nueva carrera al atributo carreras. En caso de error, se mostrará mediante un MessageBox.
  - e. Al presionar el botón Resultados se deberá mostrar por pantalla las detalle guardada por carreras.
6. La interface IRespuesta sólo contará con el método RespuestaCarrera que recibirá un atributo genérico id.
7. Clase InfoCarrera:
- a. Atributo estático randomizer. El mismo se inicializará en un constructor de clase.
  - b. El Thread carrera ejecutará el método Correr.
  - c. Correr frenará el código durante un tiempo aleatorio de entre 1 y 5 segundos. Luego de transcurrir este tiempo, utilizará el método RespuestaCarrera de callback pasando como parámetro el atributo id.
8. Clase LasCarreras:
- a. El atributo id deberá inicializar el 0.

- b. La propiedad Detalle utilizará el set para generar un archivo en el escritorio de la máquina donde se ejecute llamado "carreras.txt". El get retornará el contenido del mismo archivo.
- c. Método de clase AgregarCarrera hará los siguientes pasos, en el siguiente orden:
  - i. Incrementará id.
  - ii. creará un nuevo InfoCarrera y lo agregará a misCarreras.
- d. RespuestaCarrera hará los siguientes pasos, en el siguiente orden:
  - i. Creará un mensaje con el siguiente formato: "Terminó el hilo {0}."
  - ii. Se guardará en SQL el id de la carrera y la hora en que terminó en la tabla Carreras (ConnectionString: "Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist Security Info=False;Initial Catalog=Carrera;Data Source=.\sqlExpress").
  - iii. Guardará el mensaje en la bitácora.
  - iv. Ejecutará el evento AvisoFin.
- e. El operador + lanzará la excepción CantidadInvalidaException en el caso de que la cantidad sea menor a 1.
- f. Si cantidad es mayor a 0, deberá agregar tantos hilos cómo indique dicha cantidad