

**Universidad Tecnológica Nacional**  
**Facultad Regional Avellaneda**



**Técnico Universitario en Programación**

**Técnico Universitario en Sistemas  
Informáticos**

**Materia:** Laboratorio de computación I

Apellido:		Fecha:								
Nombre:		Docente:	<b>Rampi, Mario Scarafilo, Germán</b>							
División:		Nota:								
Legajo:		Firma:								
Instancia	PP		RPP		SP	X	RSP		FIN	

Parte 1: Consigna completa del primer parcial

Parte 2: Aplicar los temas vistos en la cátedra:

**Excepciones:**

- Todo tipo de excepción debe ser controlada. No debería haber ningún caso en el que el programa se rompa abruptamente.
- Por otro lado se deberá crear mínimo una excepción propia. La misma debe responder a una lógica determinada, según el proyecto que estén desarrollando (Por ejemplo que encapsule algún método que desarrolle alguna lógica, o que resuelva algo particular de su proyecto).

**Archivos y serialización:**

- Deberán guardar en un archivo txt, un log de errores, documentando fecha y hora del incidente, descripción, clase y método en donde ocurrió.
- El programa deberá contar con una opción de configuraciones que serán leídas desde un archivo Json. Cualquier cambio en estas configuraciones deben ser persistidas en el mismo archivo. Por ejemplo: imágenes, colores, fuentes, sonidos, etc.
- Una opción del programa permitirá guardar en xml algún dato específico generado por su programa: por ejemplo un informe diario de las ventas.
- Cualquier otra opción de serialización será bienvenida: por ejemplo un informe generado en pdf.

**Generics:**

- Aplicar el concepto de Generics en donde se considere necesario.

**Interfaces:**

- Implementar un mínimo de tres interfaces. La primera (y la típica) es interfaces y serialización. El resto deberá adaptarse a las necesidades de su proyecto.

**Test Unitario:**

- Realizar un mínimo de 3 métodos de test unitario. Los métodos que se prueban deben tener cierto grado de complejidad (por ejemplo no aplicaría si quiero probar un método que calcula un promedio).

**Base de datos:**

- Aplicar consultas dml para manipular los datos de su programa. El mismo debe estar preparado para pegarle a los dos motores de Base de datos que vimos en la cursada (Sql Server y FireBase). Los métodos correspondientes a las conexiones a cada motor, tienen que estar debidamente separados en clases. Deberán agregar una opción en el perfil de administrador que permita elegir con cuál motor se va a trabajar.

**Delegados:**

- Aplicar delegados a distintos métodos de ordenamiento utilizando expresiones lambda.
- Crear un delegado de tipo propio.
- Aplicar tipos de delegados.
- Envío de datos entre formularios mediante delegados.

**Eventos:**

- Crear un mínimo de 2 eventos propios que deben ser disparados y capturados en distintas situaciones.
- Manejar eventos predeterminados (ej el evento click de un botón) de manera dinámica (Event Handler)

**Hilos:**

- Aplicar un mínimo de dos tareas asincrónicas (no incluye la conexión a base de datos Firebase). Un ejemplo podría ser permitir la cancelación de la carga de un archivo muy pesado. Otro ejemplo: realizar la carga de un archivo mediante un hilo.