

**Universidad Tecnológica Nacional  
Facultad Regional Avellaneda**



Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos

Materia: LABORATORIO III

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

1-(3pts.) Crear un **delegado**, llamado **MiDelegado**, para procedimientos que reciban como parámetro un objeto de tipo **Control**.

Dentro del formulario de inicio (que poseerá un Button y un TextBox) se debe crear un atributo de tipo **MiDelegado** y crear dos métodos:

- El primero que reciba un **Control** y muestre por medio de un **MessageBox** si es un **TextBox** o un **Button** (en caso de no ser ninguno de los dos, lanzar una excepción que informará lo acontecido).
- El segundo método, mostrará (por **MessageBox**) el nombre del **Control**.

En el evento Load del formulario de inicio, instanciar al **único** delegado, **combinarle** los dos métodos e invocarlo.

2-(4pts.) Crear la clase **Numero**, que contenga una lista genérica de tipo **Int32**, con una propiedad "**Cantidad**" que sólo permita asignar un valor al atributo "**\_cantidad**" y un evento (diseñarlo para que reciba un **Object** y un **EventArgs**, su retorno será **void**). Si el valor que se intenta asignar es cero, se deberá lanzar una excepción de tipo **ArgumentException** informando de lo acontecido. Si el valor es positivo, dejar asignarlo. Si el valor es negativo disparar el evento **EsNegativo** cuyo manejador tendrá que escribir en un archivo de texto (log.txt) la fecha (hh:mm:ss) y el valor, asignarlo, y en el formulario de inicio se mostrará por **MessageBox** el valor negativo.

3-(4pts.) Teniendo la tabla **Proveedores**, el DataSet **dsStock** (que tendrá cargado los DataTables **dtDomicilios** y **dtProveedores**) y el DataAdapter **daProveedores**: Configurar el **SelectCommand** y el **UpdateCommand**, enunciando los comandos y cada uno de los parámetros (que se deberán nombrar como **@p1**, **@p2**, etc).

Impactar los cambios en el evento **formClosing** del formulario de inicio.

Respetando los datos de los siguientes gráficos:

Tabla - dbo.Proveedores			
Nombre de columna	Tipo de datos	Permitir valores nulos	
codProveedor	int	<input type="checkbox"/>	
razonSocial	varchar(50)	<input checked="" type="checkbox"/>	
cuit	varchar(13)	<input checked="" type="checkbox"/>	
idDomicilio	int	<input checked="" type="checkbox"/>	

4-(3pts.) Generar un formulario que se encargue de cargar los DataTables **dtDomicilios** y **dtProveedores**, generando la relación entre ellos. El formulario tendrá un control visualizador y un Button. En el manejador del botón, utilizar el método **Select** del DataTable **dtDomicilios** para obtener a los proveedores que viven en QUILMES. Mostrar **todos** los datos de los proveedores.

Domicilios	Proveedores
CodigoDomicilio	CodigoProveedor
CalleDomicilio	RazonSocialProveedor
NumeroDomicilio	CuitProveedor
LocalidadDomicilio	CodigoDomicilio

**TIEMPO MAXIMO PARA RESOLVER EL EXAMEN 60 MINUTOS.**