

**Universidad Tecnológica Nacional
Facultad Regional Avellaneda**



Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos

Materia: Laboratorio de Programación II

Apellido:		Fecha:	12/07/2018
Nombre:		Docente ⁽²⁾ :	F. Dávila / H. Dillon
División:		Nota ⁽²⁾ :	
Legajo:		Firma ⁽²⁾ :	
Instancia ⁽¹⁾ :	<input type="checkbox"/> PP <input type="checkbox"/> RPP <input type="checkbox"/> SP <input type="checkbox"/> RSP <input type="checkbox"/> FIN <input checked="" type="checkbox"/> 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.**
- **Cada tema vale 1 (un) punto (Herencia, Generics, Test Unitarios, etc.). La correcta documentación también será evaluada.**
- **Se deberán tener al menos el 60% de los temas de cada parcial bien para lograr la aprobación.**
- **Errores de conceptos de POO anulan el punto.**
- Colocar sus datos personales en el nombre del proyecto principal, colocando: Apellido.Nombre.AñoCursada. Ej: Pérez.Juan.2018. 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.

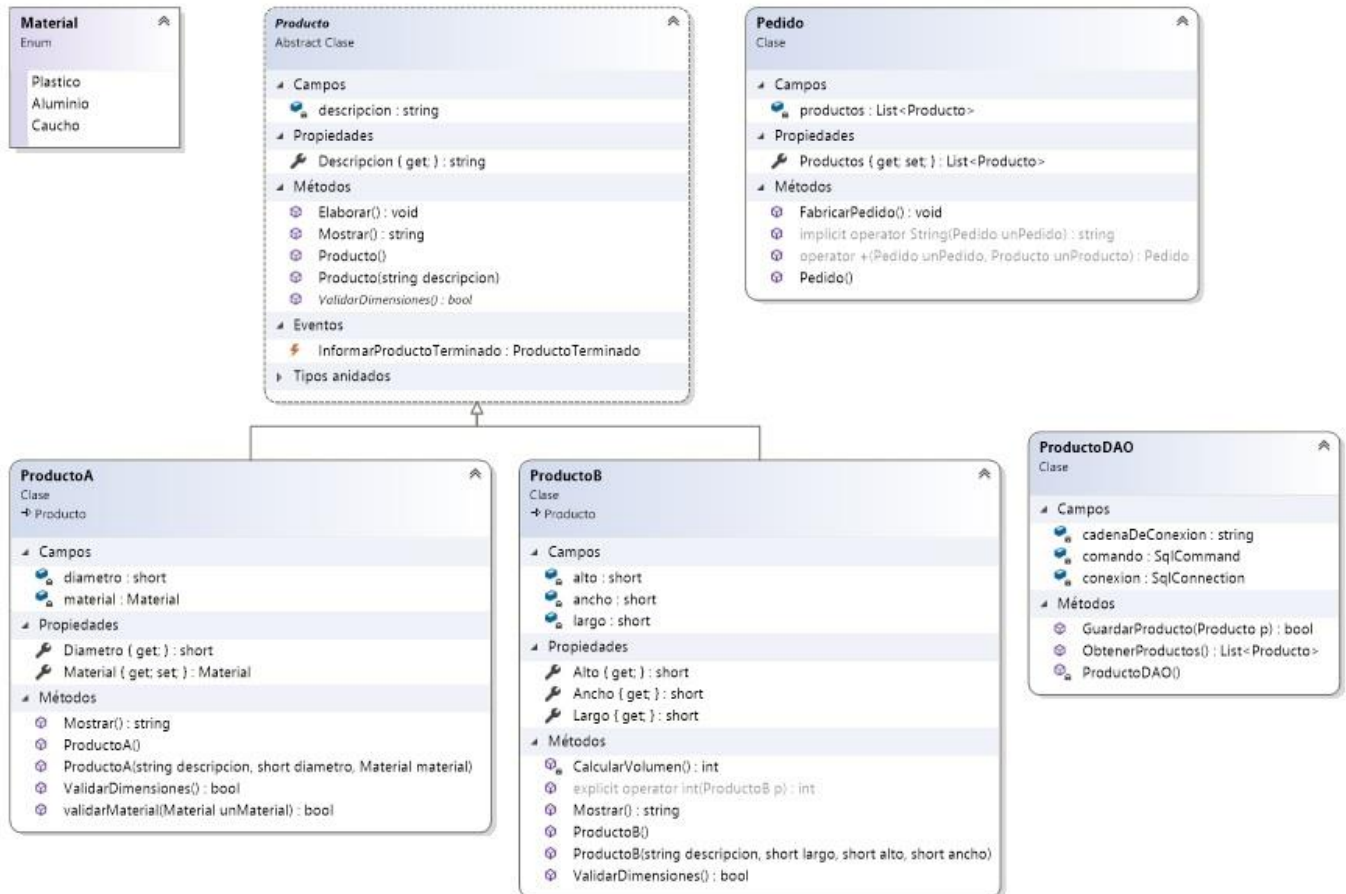
Al finalizar, colocar la carpeta de la Solución completa en un archivo ZIP que deberá tener como nombre

Apellido.Nombre.AñoCursada.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**. Luego retirarse del aula y aguardar por la corrección.

TIEMPO MÁXIMO PARA RESOLVER EL EXAMEN 100 MINUTOS.

El examen consiste en una simulación de solicitudes de pedidos para la fabricación de productos.

- 1) Dentro de la Biblioteca de clases llamada Entidades agregar el contenido correspondiente a las siguientes clases:



- a. Clase ProductoDAO -> **Sólo segundo parcial y final**
 - i. Crear una nueva base de datos con el nombre **final-20180712** y crear la tabla Productos ejecutando el archivo .sql provisto.
 - ii. **Tener en cuenta:**
 Al insertar un "ProductoA" los campos correspondientes a un "ProductoB" quedarán en null y viceversa.
 En el campo "tipo" insertar "A" para instancias de ProductoA y "B" para instancias de "ProductoB".
- b. Clase Producto
 - i. **Sólo primer parcial:** ValidarDimensiones no tendrá implementación en la clase Producto.
 - ii. Mostrar en producto tendrá implementación y podrá ser sobrescrito en las clases hijas.
 - iii. **Sólo segundo parcial y final:** Crear un delegado llamado ProductoTerminado con la siguiente firma "void ProductoTerminado(Object sender, EventArgs e)" y un evento llamado InformarProductoTerminado cuyo tipo será el delegado antes mencionado.
 - iv. **Sólo segundo parcial y final:** El método Elaborar deberá guardar en la base de datos el producto y disparar el evento InformarProductoTerminado informando el producto elaborado.
- c. Clase ProductoA:
 - i. Mostrar agregará los datos propios.
 - ii. **Sólo primer parcial y final:** ValidarDimensiones retornará true si el diámetro del producto es un valor "par" y se encuentra en el rango de los 30 y 50 cm.
 - iii. **Sólo segundo parcial y final:** ValidarMaterial será utilizado en el set de la propiedad Material. Retornará true si el diámetro del producto no supera los 10cm para el caso de aluminio, 15cm para el caso de caucho y cualquier medida para plástico. Caso contrario lanzará una excepción del tipo MaterialException con el mensaje "No se puede fabricar una pieza de {material} y diámetro de {diámetro} centímetros." (Agregar un nuevo proyecto "Excepciones" a la solución y crear dicha excepción).
- d. Clase ProductoB:
 - i. **Sólo primer parcial y final:** ValidarDimensiones retornará true si la suma del largo, ancho y alto no supera los 100 cm.

- ii. CalcularVolumen retornará el volumen obtenido de la multiplicación entre el alto, largo y ancho.
- iii. **Sólo primer parcial:** Sobrecargar el operador explicit donde se retornará el volumen del producto.
- iv. Mostrar retornará una cadena con todos los datos del producto incluyendo el volumen. (**Sólo primer parcial:** No podrá invocar directamente a CalcularVolumen dentro de Mostrar).

e. Clase Pedido:

- i. **Sólo primer parcial:** Sobrecargar el operador implicit el cual deberá listar y retornar un string con los datos de todos los productos incluidos en el pedido.
- ii. El operador + agregara un producto al pedido teniendo en cuenta que cada pedido puede contener un máximo de 5 productos. (**Sólo primer parcial y final:** Adicionalmente validar las dimensiones del producto antes de agregarlo al pedido).
- iii. **Sólo segundo parcial y final:** FabricarPedido recorrerá la lista de productos, por cada uno simular un retardo de 1 segundo y luego elaborar el producto.

2) Quienes estén haciendo el **primer parcial**:

- a. Crear dentro de la solución un nuevo proyecto de consola llamado “RecuperatorioPP”, crear un pedido, al menos 6 productos (3 de cada tipo), agregarlos al pedido y mostrar el mismo por pantalla. Probar todos los métodos y operadores creados.

FIN PRIMER PARCIAL

3) Dentro del formulario FormPedido ubicado en el proyecto VistaForm se deberá:

- a. En el manejador del evento Click del boton ConfirmarPedido se deberá instanciar el hilo del formulario y sobre él iniciar la fabricación del pedido.
- b. Asegurarse de matar el hilo del formulario antes de cerrar el mismo en caso de que siga vivo.
- c. En los manejadores btnAgregarA_Click y btnAgregarB_Click debajo del comentario “//Código alumno” Agregar el código necesario para :
 - i. Instanciar un producto partiendo de los datos ingresados en el formulario.
 - ii. Asociar los manejadores “TotalizarProductosTerminados” y “AgregarProductoTerminado” al evento del producto creado.
 - iii. Agregar el producto al pedido del formulario.
- d. El método CargarProductosTerminados será invocado cada vez que se inicie el formulario. Deberá recuperar de la base de datos la lista de productos terminados en pedidos anteriores y cargarlos en la lista de productos terminados. También se deberá actualizar el lblContadorProductos con la cantidad de productos recuperada.
- e. Mostrar en un MessageBox al usuario en caso de errores capturados en los métodos que corresponda.

4) Agregar un test unitario para probar que no se pueda agregar más de 5 productos a un mismo pedido.