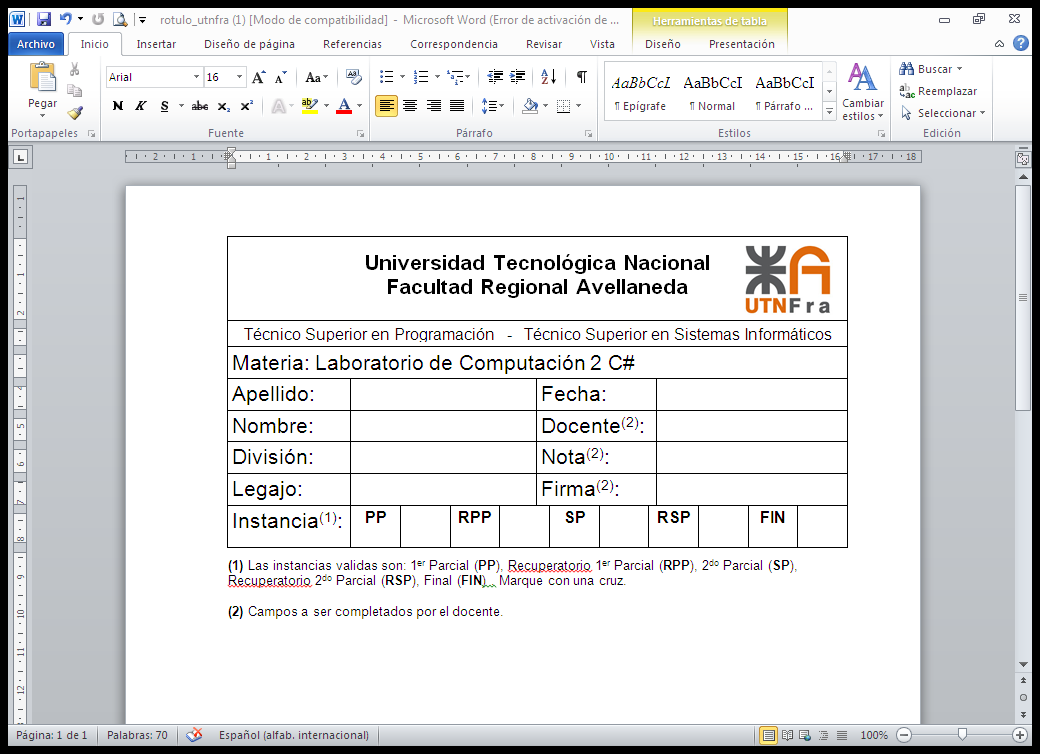
****

1. **(Final)(**1er parcial) 2pts.Realizar un método de clase que reciba tres parámetros de tipo Double (A , B y C) y que retorne el número que esta entre el menor y el mayor.
2. **(Final)**  (2pts. **)** hacer una clase que posea un constructor **estático** que inicialice un atributo, y un constructor de instancia que modifique el mismo atributo.
3. **(Final)** () 2pts.Realizar el código necesario para poder utilizar estas líneas de código en el Main

ProdImpuesto ProUno = (ProdImpuesto)33.1f;

1. **(Final)** 2pts. Realizar una interface que posea un método que retornará un objeto producto y reciba un entero y una propiedad de sólo lectura que retornará un producto.
2. **(Final)** 2pts. Realizar una clase que implemente la interface anterior y que herede de la clase abstracta con el método abstracto “Mostrar()” que no retorna valor.
3. **(Final)** (2do parcial) 2pts. Crear una clase que posea una propiedad (de sólo escritura) que solo **asigne** valor si es Positivo, de lo contrario, lanzar una excepción con el mensaje "es un números negativos".
4. **(Final)** 2pts. Realizar una estructura try-catch para la escritura de la propiedad del punto anterior que, al capturar la excepción, muestre el mensaje.
5. **(Final)** (2do parcial) 3pts. Realizar el burbujeo de una excepción **propia**, comenzando en un constructor, pasando por otro constructor y capturando por ultimo ves en el “main”.
6. **(Final)** (2do parcial) 3pts. Dada la siguiente interface:

public interface IGuardar

{ bool Serializar(); }

Realizar el siguiente método en la clase **Deposito**, que posea la siguiente firma:

public static bool GuardarXML(IGuardar productoAux)