

Enunciado:

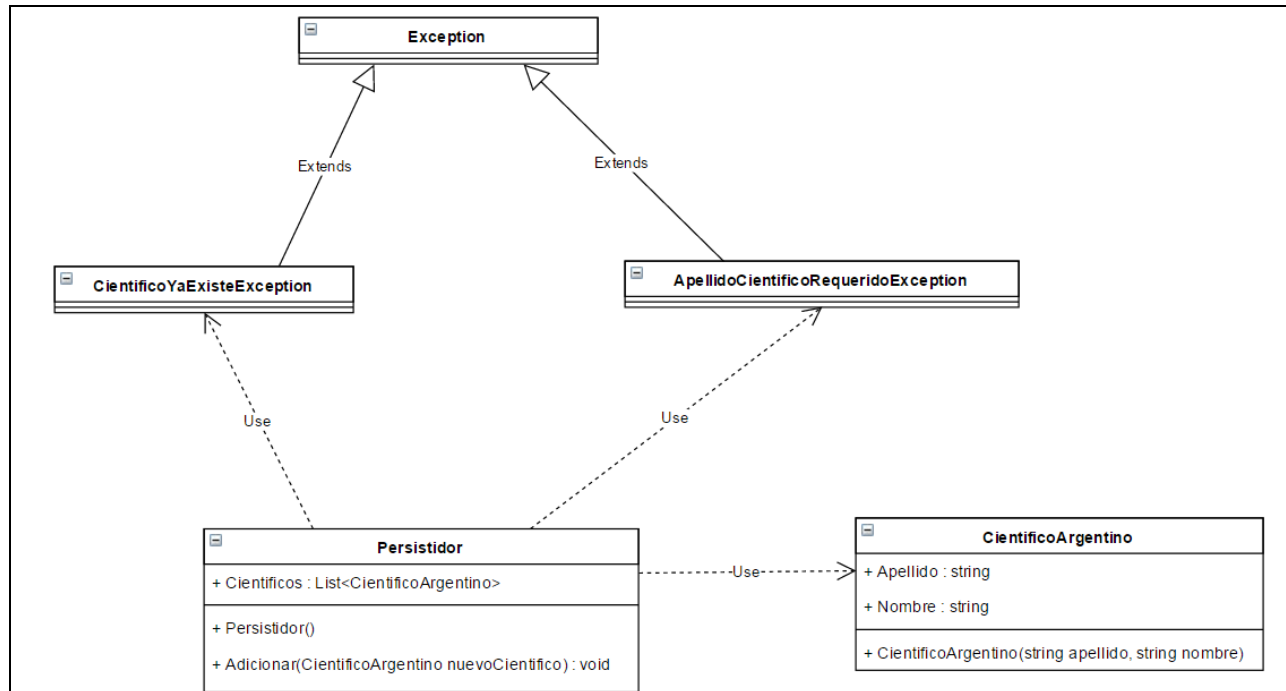
Tema: **Custom exception (excepciones personalizadas)**

Desarrolle las siguientes excepciones personalizadas, pertenecientes a la clase **Persistidor**:

Custom exception	Descripción	Lanzada por:
Persistidor.CientificoYaExisteException	Se debe disparar cuando se intenta ingresar un científico que ya existe en la lista, validando dicha existencia contra el apellido del mismo.	Disparadas por el método "Adicionar" ya que es una precondición del mismo.
Persistidor.ApellidoCientificoRequeridoException	Se debe disparar cuando se intenta ingresar en la lista de científicos uno sin apellido.	Disparadas por el método "Adicionar" ya que es una precondición del mismo.

En el diagrama de clases de la **Figura 1**, podemos ver el método público **Persistidor.Adicionar** y su firma, así como también las dos clase que representan una **excepción** personalizada (Custom exception) **"CientificoYaExisteException"** y **"ApellidoCientificoRequeridoException"**, como toda exception la clase debe heredar de **"System.Exception"**. También están presentes los constructores de las clases **"Persistidor"** y **"CientificoArgentino"**.

Figura 1



Método de la clase Persistidor	Descripción
<code>public void Adicionar(CientificoArgentino nuevoCientifico)</code>	Adiciona un nuevo científico a la lista de científicos existentes o en objetos seria: adiciona una nueva instancia de la clase “CientificoArgentino” a la colección del mismo tipo, dada por la propiedad de instancia <code>Persistidor.Cientificos</code> .

Figura 4	
Precondiciones sobre el método “Adicionar”	
1	Evidentemente la restricción es que en la lista de científicos no puede haber repetidos, siendo la propiedad necesaria para validar dicha restricción la propiedad apellido del científico.
2	No pueden existir científicos sin apellido.

La **Figura 2**, muestra la interfaz de usuario de la aplicación, la misma solo permite listar la colección de científicos o adicionar una nuevo a los ya existentes.

Figura 2

En el código fuente de la **Figura 3** podemos observar la rutina de atención al evento Click del único botón de la aplicación, básicamente es muy simple, solo llama al método “Adicionar” (26) el resto es un manejo de errores que se utiliza para dar aviso al usuario de que algo excepcional ha ocurrido (un error), como ser que el científico ya existe por su apellido (34,35) o que el científico no puede ingresarse debido a que su apellido es desconocido (30). En la **Figura 4**, podemos ver las restricciones que debe implementar el método “Adicionar” y que son capturadas y manejadas por el código de la **Figura 3**.

Figura 3

```

21 private void buttonAdicionar_Click(object sender, EventArgs e)
22 {
23     CientificoArgentino cientifico = new CientificoArgentino(textBoxApellido.Text, textBoxNombre.Text);
24     try
25     {
26         _Persistidor.Adicionar(cientifico);
27     }
28     catch (Persistidor.ApellidoCientificoRequeridoException)
29     {
30         MessageBox.Show("El apellido es de ingreso obligatorio");
31     }
32     catch (Persistidor.CientificoYaExisteException)
33     {
34         string mensaje = string.Format("El científico {0} ya existe.", cientifico.Apellido);
35         MessageBox.Show(mensaje);
36     }
37 }

```