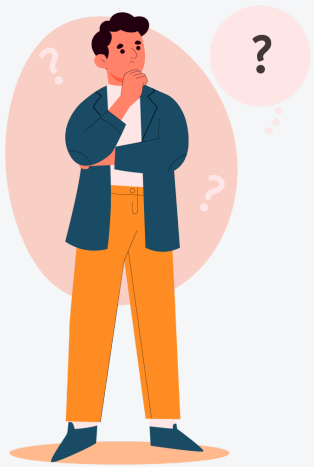
Tratamiento de Clases en Entorno Visual

Sitio: Agencia de Habilidades para el Futuro
Curso: Desarrollo de Sistemas Orientado a Objetos 1° D
Libro: Tratamiento de Clases en Entorno Visual

Imprimido por: Eduardo Moreno
Día: lunes, 31 de marzo de 2025, 00:03

Tabla de contenidos

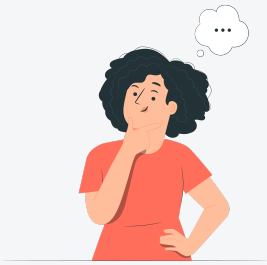
Preguntas orientadoras



- ¿Cómo creamos una clase en un entorno visual ?
- Y para agregar a una solución una clase, ¿se requiere solo los controles?
- ¿Refactorizar es un término asociado a las clase?
- ¿El formulario es una clase? ¿Y qué define el método utilizado?

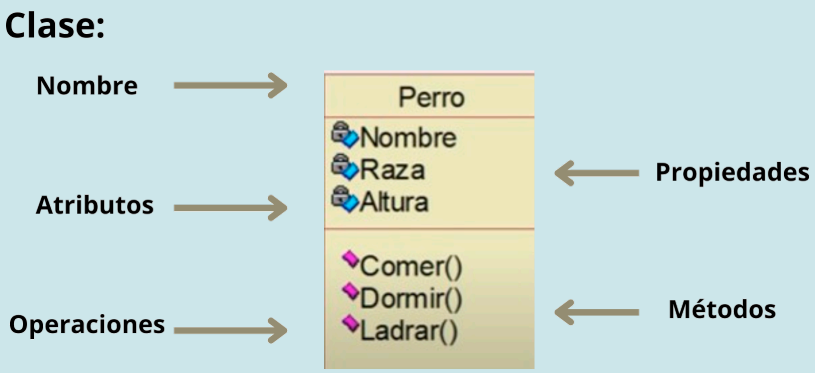
Recordando conceptos

Retomemos algunos temas de la unidad anterior. Podemos decir que las clases son plantillas para definir elementos (objetos) y estas pueden estar directamente relacionadas unas con otras.



Los objetos son elementos con comportamiento y estado, con métodos y atributos concretos. En otras palabras, son instancias de clase e interactúan por medio de mensajes.

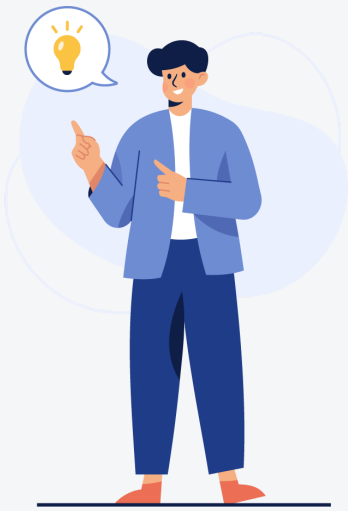
Ejemplo: Tenemos la clase *Perro*, donde la parte superior muestra los atributos y la parte inferior a los métodos (acciones).



Ya refrescamos estos conceptos. ¿Y ahora? El paso siguiente será trabajar la clase en el entorno visual. A continuación, pasemos entonces a ver cómo creamos en el entorno visual a la clase desde la solución.

Creación de una clase desde la solución

Antes de arrancar, ¡algunos #tips! Mientras estamos programando vamos a considerar alguna de las siguientes palabras public, private, protected. Están siempre presentes antes de comenzar alguna línea de código.



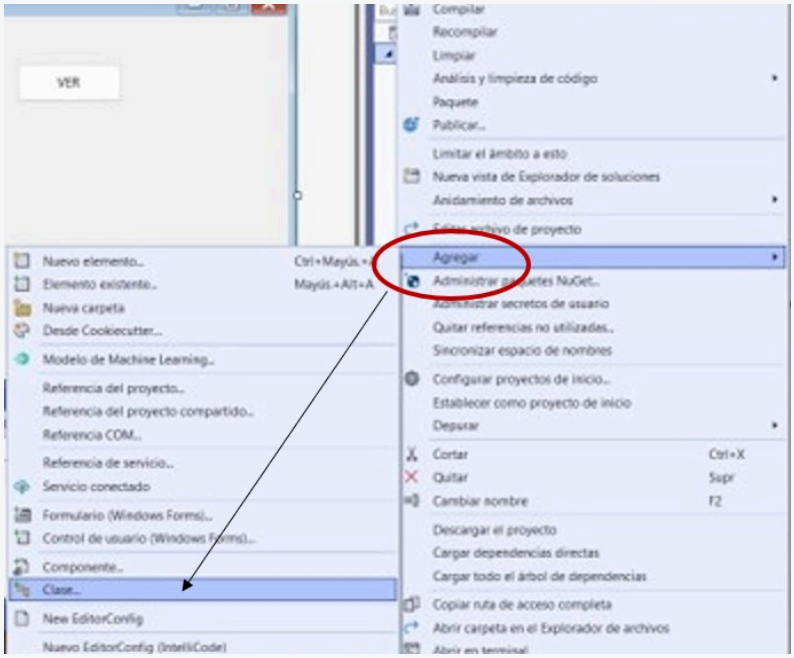
¿Recordas que significan? Para que no tengas que buscar en el material de estudio rápido hacemos un repaso.

- Public se puede acceder desde cualquier lugar.
- Private sólo se puede acceder desde la propia clase.
- Protected sólo se puede acceder desde la propia clase o desde una clase que herede de ella.

Creando la clase desde la solución

Ahora sí, continuemos. Estos conceptos mencionados pertenecen al encapsulamiento. Pero, ¿cómo creamos la clase desde la solución?

1) Con botón derecho sobre el nombre de la solución seleccionamos en la ventana que aparece la opción agregar y luego buscamos clase.



2) En la plantilla de elemento seleccionamos clase y le colocamos un nombre, con la aceptación este nuevo elemento pasa a ser parte de nuestra solución.

A continuación, pasemos a programar la clase. En otras palabras, ¡vayamos a codificar!

Codificar la clase

Lo primero que vamos a definir es la visibilidad de nuestra clase. Vamos a tratarla como **public**

Para lograrlo, armamos el esqueleto de los atributos y métodos, si es necesario el método puede pasar a ser una función y retornar un resultado.

```
6
7   namespace CONTENIDOS  /* nombre de la solución */
8   {
        0 referencias
9       public class Perro  /* CLASE creada como pública */
10      {
11          /* ATRIBUTOS
12           * *************  */
13
14          public string? nombre;
15          public string? raza;
16          public double altura;
17
18          /*      METODOS
19           comienzan en mayuscula
20          * *************  */
            0 referencias
21          void Comer()
22          {
23          }
            0 referencias
24          void Dormir()
25          {
26          }
            0 referencias
27          void Ladrar()
28          {
29          }
30      }
31
32  }
```

Ahora en el formulario, en alguno de los controles convertidos en objetos (clases ya instanciadas) codificamos algún evento.

En esa codificación vamos a usar la clase recién creada. Lo primero debemos es instanciar un objeto y para el formato de escritura es el siguiente:

Tipo identificador=New Tipo();

- Tipo es la **clase**
- Identificador es el nombre del **objeto** de esa clase.

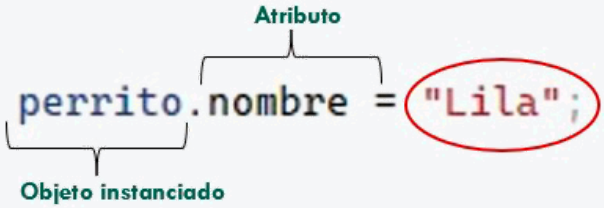
La codificación se escribe en el evento click de un button.

```
private void btnUno_Click(object sender, EventArgs e)
{
    // Instanciamos un objeto de la clase Perro
    // este objeto lo llamamos perrito

     Perro perrito = new Perro();

}
```

Para hacer referencia a los atributos de la clase se escribe el nombre del objeto a continuación un punto y sin espacios el nombre del atributo. En este ejemplo le asignamos el valor “Lila”.



¡Más #tips! Para que los métodos que describimos en la clase puedan ser visibilizados en el formulario se deben declarar como públicos.

Cuando trabajamos dentro de los métodos y se hace referencia a algún atributo de la clase usamos la palabra **this** seguida de un punto y a continuación el atributo en cuestión.

```
this.atributo
```

Ampliemos este punto que es muy importante analizando la codificación de los métodos.

Codificar los métodos

¿Por qué nos detenemos en este punto? Los métodos definen el comportamiento de los objetos de una clase. La implementación se suele ocultar al exterior de la clase.

Existen distintos métodos. Veamos algunos:

- Constructor: sirve para inicializar un objeto al crearlo. Existe sobrecarga (distintos parámetros). Coincide con el nombre de la clase y no devuelve nada por definición

```
public Perro( string nombre, string raza, string altura)
{
        this.nombre = nombre;
        this.raza = raza;
        this.altura = altura;
}
```

parámetros

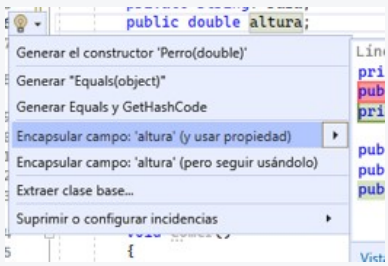
- Get y Set: Sirven para obtener o para modificar los atributos de una clase.

```
public string Nombre
{
    get{return nombre ;}
    set{nombre = value ;}
}
```

Encapsulando atributos

Observamos que el método está vinculado a una propiedad. En este sentido, volviendo a la clase que estuvimos programando vamos a encapsular los tres atributos (nombre, raza y altura) para crear los métodos **get** y **set**

Nos paramos con botón derecho sobre el atributo y seleccionamos acciones rápidas y refactorizaciones, dentro seleccionamos:



Al crearla, el atributo pasa de ser público a privado de forma automática y el método para acceder es público.

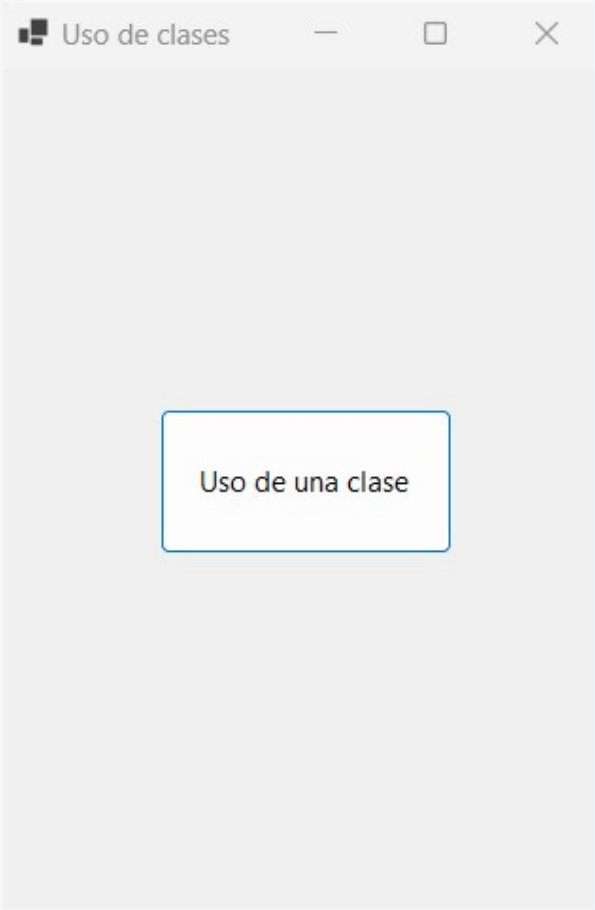
¿Te parece que veamos algunos ejemplos? ¡Avancemos!

Ejemplo de aplicación

¡Relacionemos lo aprendido!

Retomemos el ejemplo utilizado en la introducción del módulo.

Diseño de la solución:



Podés descargar la codificación de la clase y del button ***uso de una clase*** desde la carpeta Soluciones del entorno visual.

Las pantallas de ejecución son las siguientes:


Uso de clases
Carga de la clase
ingrese nombre del Perro
Aceptar
Cancelar
PELUSA
Uso de clases
Carga de la clase
ingrese raza del Perro
Aceptar
Cancelar
LABRADOR
Ingreso de los datos con cuadro de diálogo
Uso de clases
Carga de la clase
ingrese altura del Perro
Aceptar
Cancelar
0.70
Salida (cuadro de diálogo) y observa el método set
Uso de clases
salida de datos
El perrito se llama PELUSA de raza LABRADOR y su altura 0.70 cm
Aceptar

En resumen

Vimos como tratar clases en el entorno visual cuando ellas no son elementos de la barra de herramientas.

Es fundamental agregarlas a la "solución" que contiene al formulario.

A través de las líneas de código le dimos vida a los atributos y métodos.

¡Sigamos avanzando!