



## Laboratorio No. 09

# Métodos y parámetros

## Objetivos



Definir clases y sus elementos principales: atributos y métodos.



Utilización de los principales tipos de datos para atributos y/o métodos de una clase.

## Instrucciones



Codifique la solución de cada uno de los problemas en C#.



Comprima la carpeta en un formato `.zip` o `.rar` y colóquelo en el portal académico en el respectivo recurso.



## Práctica en Laboratorio

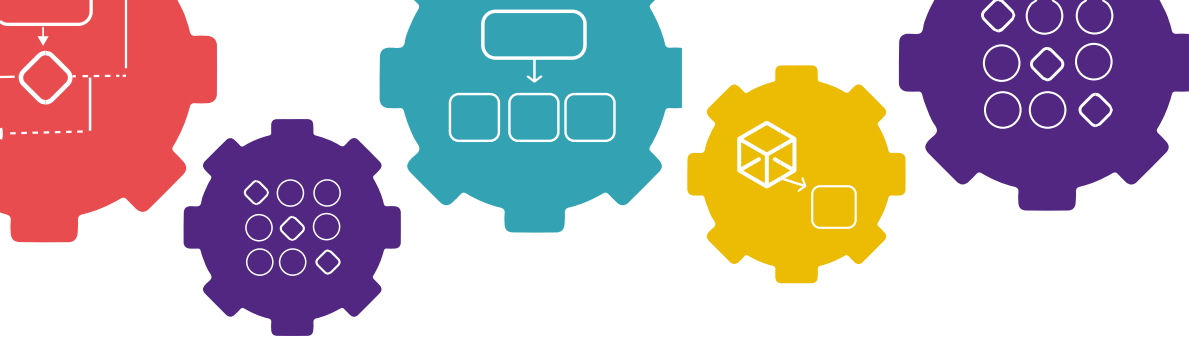
1.

Ingresa a Visual Studio Code y cree una nueva **aplicación de Consola** para Windows en C#, nombre el proyecto **L9+<iniciales>+<carne>**.

2.

Agregue una nueva clase a su proyecto:

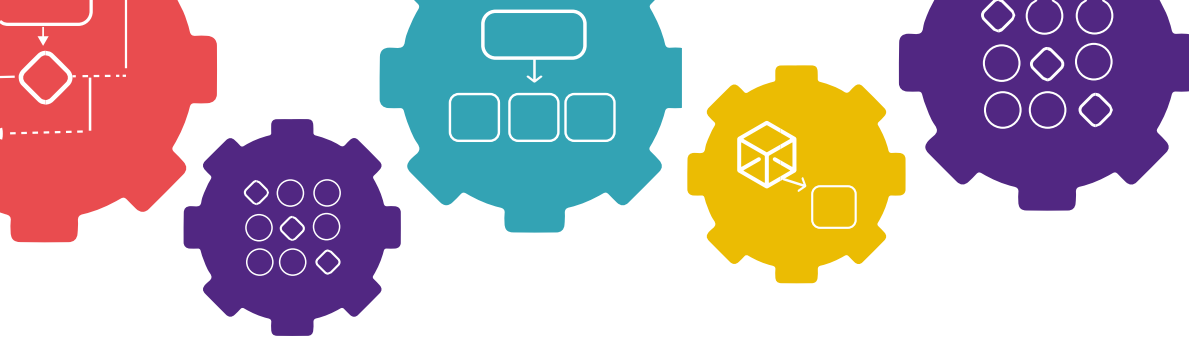
- El nombre de la clase será: **Automovil** (sin tilde)
- Defina los siguientes atributos *privados*:
  - **modelo**: tipo *int*. Guarda el modelo del automóvil. Por defecto tendrá el valor de 2019, establecido en el constructor.
  - **precio**: tipo *double*. Almacena el precio del automóvil en quetzales. Por defecto el valor será 10000.00, establecido en el constructor.
  - **marca**: tipo *string*. Almacena la marca del automóvil. Por defecto tendrá un texto vacío "", establecido en el constructor.
  - **disponible**: tipo *bool*. Indica si el auto se encuentra disponible o no. Por defecto el valor será *false*, establecido en el constructor.
  - **tipoCambioDolar**: tipo *double*. Tipo de cambio de dólares a quetzales. Por defecto será de 7.50, establecido en el constructor.
  - **descuentoAplicado**: tipo *double*. Si se le aplicó algún descuento al precio, por defecto será 0.00, establecido en el constructor.



### 3. Defina los siguientes métodos *públicos*:

- **Automovil:** constructor. Establece los atributos por defecto listados anteriormente
- **DefinirModelo:** procedimiento (void). Recibe un parámetro entero llamado *unModelo* que define el (atributo) modelo del automóvil.
- **DefinirPrecio:** procedimiento que recibe un parámetro *double* llamado *unPrecio* que define el precio del automóvil.
- **DefinirMarca:** procedimiento que recibe un parámetro *string* llamado *unaMarca* que define la marca del automóvil.
- **DefinirTipoCambio:** procedimiento que recibe un parámetro *double* *unTipoCambio* que define el tipo de cambio.
- **CambiarDisponibilidad:** procedimiento. Si el atributo **disponible** es verdadero, lo cambia a falso; si es falso, lo cambia a verdadero.
- **MostrarDisponibilidad:** función tipo *string*. Si el atributo **disponible** es verdadero devuelve el texto: "Disponible"; si es falso devuelve el texto: "No se encuentra disponible actualmente"
- **MostrarInformacion:** función tipo *string*. Devolverá un texto con los datos de la siguiente manera (hacer las conversiones pertinentes):  
"Marca: " + [marca] + ". Modelo: " + [modelo] + ". Precio de venta: Q" + [precio] + ". Precio en dólares \$" + [calcular precio en dólares] + ". " + **MostrarDisponibilidad()**.
- **AplicarDescuento:** procedimiento. Recibe como parámetro un *double* llamado *miDescuento*. Le asigna el valor de *miDescuento* a la variable *descuentoAplicado* y modifica el precio del vehículo con el descuento aplicado (Realizar la operación aritmética y luego llamar al procedimiento *DefinirPrecio* para guardar el nuevo valor).

### 4. En el método principal del programa; declare un objeto del tipo **Automovil** llamado "objAutomovil". (No declare ninguna otra variable en el programa principal).



5. Genere el código necesario, que permita el ingreso de los campos *modelo*, *precio*, *marca* y *tipo de cambio*, por medio de *consola*.

6. Guardará los datos recabados en el objeto `objAutomovil` y se mostrarán los mismos en la pantalla. Luego debe de preguntar si "Desea cambiar disponibilidad (Disponibilidad actual: <disponibilidad>)" y codifique la entrada del usuario para editar la disponibilidad, finalmente imprima el resultado. el campo para añadir un descuento y el botón de "Aplicar descuento".

7. Solicite al usuario el monto de descuento a aplicar un y aplíquelo al vehículo.

8. Muestre la información del automóvil en consola.