**Tarea: Fundamentos de C#**

Jennifer Rebeca García Moncada

Universidad Politécnica Internacional

Programación III

Profesor: Alexander Benjamin Curling

Setiembre, 2024

**Investigación Teórica**

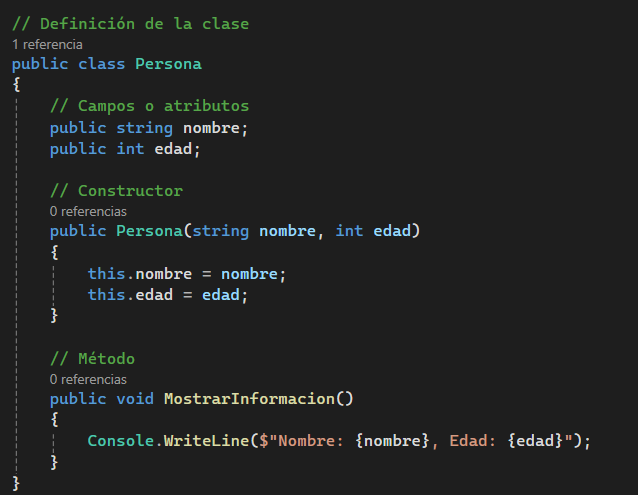
Investiga los siguientes conceptos de C#:

1. ¿Qué es una clase en C# y cómo se crea?

Una plantilla o modelo que se utiliza para crear objetos en C# se conoce como clase. Define los campos y métodos que puede tener un objeto creado a partir de la clase. Una clase es en esencia una representación de un concepto en el código que permite estructurar de manera coherente la lógica y los datos.

Para crear una clase en C#, se utiliza la palabra clave class seguida del nombre de la clase.

Ejemplo:



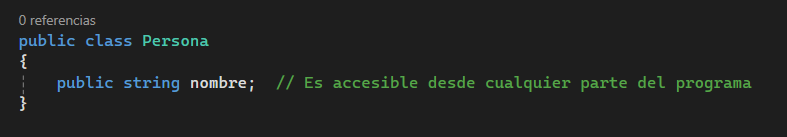
1. ¿Cuál es la diferencia entre los delimitadores de acceso private y public?

La principal distinción entre los delimitadores de acceso privado y público en C# radica en el nivel de accesibilidad que brindan a los miembros de una clase o estructura (atributos y métodos).

Public: Los miembros que se definen como público pueden acceder al programa desde cualquier lugar. Cualquier otra clase, método o incluso código externo puede acceder y usar estos miembros.

La exposición de una propiedad, método o campo para que sea visible y accesible a cualquier otra parte del programa es su uso típico.

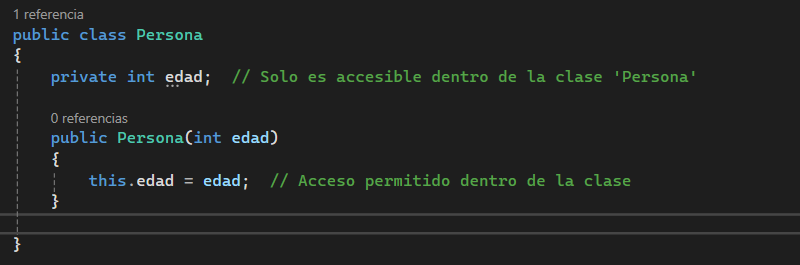
Ejemplo:



Private: Solo los miembros privados pueden acceder dentro de la misma clase o estructura. Ninguna otra clase o código puede acceder directamente a los miembros privados.

Se utiliza con frecuencia para encapsular métodos y datos que no se pueden acceder o cambiar directamente desde fuera de la clase. Esta es una manera de salvaguardar la integridad de los datos.

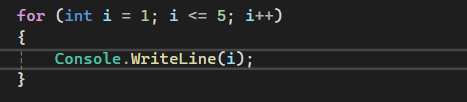
Ejemplo:



1. ¿Cómo funcionan los ciclos for y while en C#?

Los ciclos for y while son estructuras de control utilizadas en C# que permiten repetir un bloque de código en función de una condición.

**Ciclo For:** Cuando sabes cuántas veces quieres repetir una operación, el ciclo for es útil. La inicialización, la condición y la actualización son sus tres componentes principales. Su estructura es:



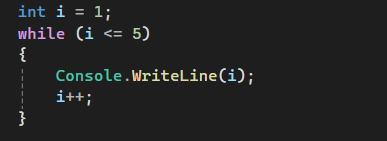
**Características:**

Se ejecuta al inicio del ciclo. Se utiliza con frecuencia para especificar una variable de control.

Antes de cada iteración, se evalúa una expresión booleana conocida como condición. El ciclo seguirá si es cierto.

Al final de cada iteración, se ejecuta una actualización. El uso más común es para ajustar la variable de control.

**Ciclo While:** El ciclo while ejecuta un bloque de código una y otra vez mientras se mantiene una condición booleana verdadera. Su estructura es la siguiente:

****

**Diferencias clave entre for y while:**

Ciclo for: Se usa cuando conoces de antemano cuántas veces se repetirá el ciclo.

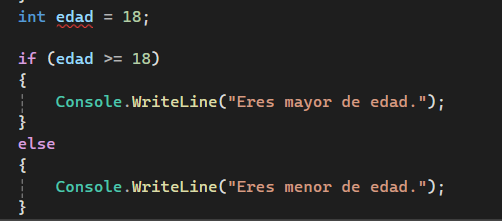
Ciclo while: Se usa cuando no necesariamente conoces la cantidad de iteraciones, y deseas que el ciclo continúe hasta que una condición deje de ser verdadera.

Ambos ciclos pueden ser interrumpidos antes de completar todas las iteraciones usando la instrucción break, o pueden continuar a la siguiente iteración con continue.

1. Explica el uso de la estructura if-else en C#.

En C#, las decisiones sobre el flujo de ejecución de un programa se toman utilizando la estructura if-else. Permite la ejecución de varios bloques de código según si una condición es verdadera o falsa.

Ejemplo:

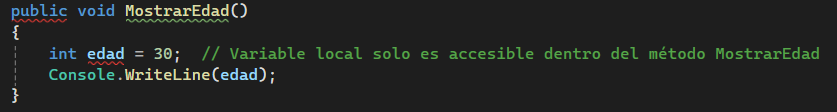


if: Evalúa una condición booleana. Si es true (verdadera), ejecuta el bloque de código dentro de las llaves.

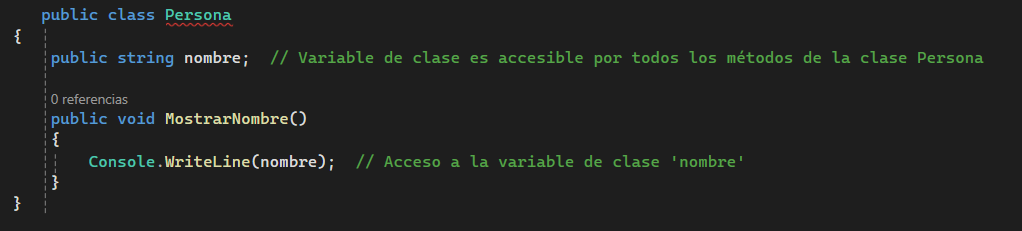
else: Se ejecuta si la condición en el if es false (falsa).

1. ¿Cuál es la diferencia entre variables locales y variables de clase en C#?

**Variables Locales:** Son variables que se declaran dentro de un método, constructor o bloque de código y solo son accesibles dentro de ese mismo método o bloque.



**Variables de Clase:** Son variables que se declaran directamente en una clase, fuera de cualquier método, constructor o bloque. Estas variables representan el estado o atributos de una clase y son accesibles en cualquier método de la clase, dependiendo de su nivel de acceso (public, private, etc.).



**Diferencias**

|  |  |  |
| --- | --- | --- |
| Característica | Variable Local | Variable de Clase |
| Lugar de declaración | **Dentro de métodos, constructores o bloques.** | **Fuera de métodos, directamente en la clase** |
| Alcance | **Solo dentro del método o bloque se declara.** | **Toda la clase, accesible por todos sus métodos.** |
| Ciclo de vida | **Existe solo durante la ejecución del método.** | **Existe mientras la instancia de la clase este en memoria o toda la ejecución si es estática.** |
| Inicialización | **Debe ser inicialización antes de usarla** | **Tiene un valor predeterminado si no se inicializa.** |
| Visibilidad | **No puede tener modificadores de acceso.** | **Puede ser private, public, protected, etc.** |