

Parte 1: Investigación Teórica

Investiga los siguientes conceptos de C#:

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

En C#, una clase es un plano o molde que define las propiedades (atributos) y comportamientos (métodos) de un objeto.

Así se crea la clase

```
public class NombreClase {  
    // Aquí se definen las propiedades, métodos y constructores  
}
```

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

public: Accesible desde cualquier parte del código.

private: Solo accesible dentro de la clase en la que se define.

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

El ciclo `for` se utiliza cuando conoces de antemano cuántas veces quieres ejecutar un bloque de código.

- **Inicialización:** Se ejecuta solo una vez al inicio del ciclo (generalmente se inicializa un contador, como `int i = 0`).
- **Condición:** Se evalúa antes de cada iteración. Si es verdadera, el ciclo continúa; si es falsa, el ciclo se detiene.
- **Actualización:** Se ejecuta al final de cada iteración (normalmente se incrementa o decrece el contador).

El ciclo `while` se usa cuando no sabes cuántas veces se ejecutará el bloque de código, pero tienes una **condición** que determina cuándo detener el ciclo.

- **Condición:** Mientras esta condición sea verdadera, el ciclo continuará ejecutándose. Si la condición es falsa, el ciclo se detiene.

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

La estructura **if-else** en C# se utiliza para tomar decisiones en el código en función de condiciones. Dependiendo de si la condición es verdadera (`true`) o falsa (`false`), se ejecuta un bloque de código u otro.

¿Cómo funciona?

1. **if:** Evalúa la **condición** entre paréntesis. Si la condición es verdadera, se ejecuta el bloque de código dentro de las llaves `{}`.
2. **else:** Si la condición del `if` es **falsa**, se ejecuta el bloque de código dentro del `else`.

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

En C#, las **variables locales** son aquellas que se declaran dentro de un método o bloque de código, tienen un ámbito limitado a ese bloque y su vida útil termina cuando el método o bloque finaliza. No pueden llevar modificadores de acceso. Por otro lado, las **variables de clase** (o campos) se declaran fuera de los métodos, dentro de la clase, y son accesibles desde cualquier método de la misma clase. Su vida útil depende de la instancia de la clase (o de la ejecución del programa si son estáticas) y pueden llevar modificadores de acceso como `public` o `private`.