



---

# CONCEPTOS DE INTERFACES

---

Tarea en vacaciones :C



Alumno:

- Victor Manuel Aguilar Mercado

Maestra:

- Cinthia Anahí Mata Bravo

TECNOLOGIAS DE LA INFORMACION Y COMUNICACIONES

INSTITUTO TECNOLÓGICO DEL SUR DE NAYARIT



TECNOLÓGICO  
NACIONAL DE MÉXICO



## Índice

Investigación .....	2
Definición interfaz.....	2
Descripción de mi programa .....	4

## Investigación

### Definición interfaz

Una interfaz contiene las definiciones de un grupo de funcionalidades relacionadas que una clase o una estructura no abstracta deben implementar. Mediante las interfaces puede incluir, por ejemplo, un comportamiento de varios orígenes en una clase. Esta capacidad es importante en C# porque el lenguaje no admite la herencia múltiple de clases. Además, debe usar una interfaz si desea simular la herencia de estructuras, porque no pueden heredar de otra estructura o clase.

Para definir una interfaz, deberá usar la palabra clave `interface`, así como en este ejemplo:

```
interface IEquatable<T>
{
    bool Equals(T obj);
}
```

El nombre de una interfaz debe ser un nombre de identificador de C# válido. Por convención, los nombres de interfaz comienzan con una letra I mayúscula.

Cualquier clase o estructura que implementa la interfaz `IEquatable<T>` debe contener una definición para un método `Equals` que coincida con la firma que la interfaz especifica. Como resultado, puede contar con una clase que implementa `IEquatable<T>` para contener un método `Equals` con el que una instancia de la clase puede determinar si es igual a otra instancia de la misma clase.

Las interfaces pueden contener propiedades, eventos, indizadores o métodos de instancia, o bien cualquier combinación de estos cuatro tipos de miembros. Las interfaces pueden contener constructores estáticos, campos, constantes u operadores. Para obtener vínculos a ejemplos, vea Secciones relacionadas. Una interfaz no puede contener campos de instancia, constructores de instancias ni finalizadores. Los miembros de la interfaz son públicos de forma predeterminada.

Cuando una clase o estructura implementa una interfaz, la clase o estructura debe proporcionar una implementación para todos los miembros que la interfaz declara sin

proporcionar ninguna implementación predeterminada para ellos. Sin embargo, si una clase base implementa una interfaz, cualquier clase que se derive de la clase base hereda esta implementación.

Aquí se muestra este ejemplo de una implementación de la interfaz `IEquatable<T>`. La clase de implementación `Car` debe proporcionar una implementación del método `Equals`.

```
public class Car : IEquatable<Car>
{
    public string Make {get; set;}
    public string Model { get; set; }
    public string Year { get; set; }

    // Implementation of IEquatable<T> interface
    public bool Equals(Car car)
    {
        return (this.Make, this.Model, this.Year) ==
            (car.Make, car.Model, car.Year);
    }
}
```

## Descripción de mi programa.

Este programa es muy sencillo, pero para una breve explicación esta bien, este programa hace la suma de dos números sean decimales o enteros, primero empezamos con la pantalla agregando las herramientas necesarias, después le agregas una clase se pondrá las variables, en la interfaz se pondrá el procedimiento que el cual es el que ahora el trabajo y al final programamos el botón con el cual saldrá el resultado.

Espero y después pueda hacer mucho mas con una interfaz ahorita fue rápido, pero investigare más para poder hacer cosas mejores.

Lo que dice la investigación las interfaces son donde agregas los grupos o clases donde colocas el comportamiento no abstracto de ello, para poder usar una interfaz se debe de usar la palabra clave interface junto con un nombre que claramente indica tiene que ser relacionado al lenguaje de C# el cual tiene que inicializar con I y por ultimo en cualquier clase o estructura `IEquatable<T>`