**TECNOLOGICO DEL SUR DE NAYARIT**

**PROGRAMACION ORIENTADA A OBJETOS**

**CARLOS DANIEL ELIAS SOTO**

**“INTERFACES”**

**2 SEMESTRE**

**191140021**

**ITIC**

**INTERFACES EN C#**

Una interfaz contiene definiciones para un grupo de funcionalidades relacionadas que una clase no abstracta o una estructura deben implementar. Una interfaz puede definir staticmétodos, que deben tener una implementación. Una interfaz puede proporcionar una implementación predeterminada para cualquiera o todos sus miembros de instancia declarados. Una interfaz no puede declarar datos de instancia como campos, propiedades implementadas automáticamente o eventos similares a propiedades.

Mediante el uso de interfaces, puede, por ejemplo, incluir el comportamiento de múltiples fuentes en una clase. Esa 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 en realidad no pueden heredar de otra estructura o clase.

Sin embargo, la clase puede proporcionar una implementación de una interfaz solo una vez y solo si la clase declara la interfaz como parte de la definición de la clase (class ClassName: InterfaceName) Si la interfaz se hereda porque usted heredó una clase base que implementa la interfaz, la clase base proporciona la implementación de los miembros de la interfaz. Sin embargo, la clase derivada puede volver a implementar cualquier miembro de la interfaz virtual en lugar de utilizar la implementación heredada.

**RESUMEN PERSONAL**

A lo que entendí la interfaz es muy parecida a la clase papa porque los componentes que ponga en la interfaz ya sean las variables o métodos, a fuerzas deben de ir en las clases, con un componente que no se ponga en la clase te marcara error y no podrás inicializarlo.