

## EJERCICIOS DE INTERFACES

1. Define una interfaz `FiguraGeometrica` con métodos para calcular el área y el perímetro. Implementa esta interfaz en clases como `Rectangulo`, `Circulo` y `Triangulo`.
2. Define una interfaz `AccionesTelevisor` que contenga métodos para encender, apagar, cambiar de canal y ajustar el volumen de un televisor. Implementa esta interfaz en una clase `TelevisorSmart` que simule las acciones de un televisor inteligente.
3. Define una interfaz `Ordenable` con un método `comparar()` que compare dos objetos y devuelva un valor entero según el orden. Implementa esta interfaz en una clase que compare objetos de tipo `String` por longitud.
4. Crea una interfaz `OperacionMatematica` con métodos estáticos para sumar, restar, multiplicar y dividir. Implementa esta interfaz en una clase `Calculadora`.
5. Define una interfaz `Logger` con métodos para registrar mensajes de log de diferentes niveles de severidad. Implementa esta interfaz en una clase `ConsolaLogger` que imprima los mensajes por consola.
6. Crea una interfaz `Ordenable` con un método `compareTo()` que permita comparar objetos. Implementa esta interfaz en una clase `Producto` que tenga un nombre y un precio.
7. Define una interfaz `ReproductorMusical` con métodos para reproducir, pausar y detener la reproducción de música. Implementa esta interfaz en una clase `ReproductorMP3`.
8. Crea una interfaz `Formateable` con métodos para formatear texto, como `negrita()`, `cursiva()`, `subrayado()`. Implementa esta interfaz en una clase `DocumentoTexto`.
9. Define una interfaz `Conductor` con un método `conducir()` que imprima un mensaje genérico de conducción. Implementa esta interfaz en una clase `ConductorAutobus` y una clase `ConductorTaxi`.
10. Define una interfaz `Calculo` con un método privado `calcular()` y un método por defecto `imprimirResultado()`. Implementa esta interfaz en una clase `OperacionAritmetica`.
11. Crea una interfaz `Evento` con métodos para iniciar, pausar y detener eventos. Implementa esta interfaz en una clase `Reloj` que simule un reloj digital.
12. Define una interfaz `Comparador` con un método `comparar()` que compare dos objetos. Implementa esta interfaz en una clase `ComparadorStrings` que compare objetos `String` por orden alfabético.
13. Crea una interfaz `Juego` con métodos para iniciar, pausar y detener el juego. Implementa esta interfaz en una clase `JuegoAventura` que simule un juego de aventuras.
14. Define una interfaz `InterfazUsuario` con métodos para mostrar mensajes, recibir entrada del usuario y mostrar errores. Implementa esta interfaz en una clase `ConsolaInterfaz` que interactúe con la consola.
15. Diseña una aplicación que procese una lista de números enteros y realice alguna operación sobre ellos, como la suma, el producto, o la búsqueda del máximo o mínimo. Utiliza una interfaz funcional que represente esta operación y crea una clase anónima para implementarla.
16. Crea una clase `Producto` con atributos como `nombre`, `precio`, y `stock`. Luego, define una interfaz `Envable` con un método `calcularCostoEnvio()` que devuelve el costo de envío del producto. Implementa esta interfaz en las clases `ProductoElectronico` y

**ProductoFisico**. Utiliza una variable de tipo **Enviable** para calcular y mostrar el costo de envío de diferentes tipos de productos.

17. Diseña una aplicación de música con clases como **Cancion**, **Podcast**, y **Audiolibro**. Cada uno de estos elementos puede ser reproducido, pausado y detenido. Define una interfaz **Reproductor** que contenga métodos para estas acciones. Luego, utiliza una variable de tipo **Reproductor** para manejar la reproducción de diferentes tipos de contenido multimedia.
18. Implementa un sistema bancario simple con clases como **CuentaCorriente**, **CuentaDeAhorros**, y **CuentaDeInversion**. Cada una de estas clases puede realizar operaciones comunes como depósito, retiro y consulta de saldo. Define una interfaz **OperacionesBancarias** que contenga estos métodos y utilízala como atributo en las clases de cuenta para realizar operaciones bancarias.
19. Crea una aplicación para gestionar formularios en línea con clases como **FormularioRegistro**, **FormularioContacto**, y **FormularioCompra**. Cada uno de estos formularios puede ser validado y enviado. Define una interfaz **Formulario** que contenga métodos para validar y enviar el formulario. Utiliza una variable de tipo **Formulario** para manejar diferentes tipos de formularios.