

Basándose en el ejercicio de instrumentos de la semana anterior, implementar interfaces para agregar funcionalidades específicas usando ArrayList.

INTERFACES A CREAR:

A) Interface Electrico

```
public interface Electrico {  
    void encender();  
    void apagar();  
    boolean estaEncendido();  
}
```

B) Interface Afinable

```
public interface Afinable {  
    void afinarAutomaticamente();  
    boolean estaAfinado();  
}
```

NUEVAS CLASES:

C) Clase GuitarraElectrica:

Hereda de: InstrumentoCuerda

Implementa: Electrico y Afinable

Atributo adicional: encendida (boolean, private)

Override tocar(): mostrar si está encendida o apagada

Implementar métodos de ambas interfaces

D) Clase TecladoElectronico:

Hereda de: Instrumento

Implementa: Electrico

Atributos: numeroTeclas (int, private), encendido (boolean, private)

Override tocar() y mostrarInfo()

Implementar métodos de la interface

E) EN EL MAIN:

Crear ArrayList: ArrayList<Instrumento> orchestra = new ArrayList<>(); --->EL arrayList va a ser de INSTRUMENTO recuerden esto

Agregar 5 instrumentos: guitarra acústica, violín, guitarra eléctrica, flauta, teclado

Recorrer con for-each: mostrar info, tocar y afinar todos

Funcionalidades específicas usando instanceof (no todos los instrumentos se pueden afinar o encender y apagar... solo los que implementan):

```
    si en el recorrido un instrumento es instance of Electrico  
        crear una variable Electrico electrico = (Electrico) "inst que estoy viendo en  
el for..." --->Hay que castear de Instrumento a Electrico poruqe el metodo encender  
no es de instrumento....  
        electrico.encender()  
        inst.tocar
```

NOTA sobre instanceof:

El operador instanceof verifica si un objeto es una instancia de una clase o implementa una interface específica.

Retorna true o false. Es útil antes de hacer casting para evitar errores de ejecución.