

### Ejercicio:

Utilizando como referencia las siguientes clases, **modifique lo que considere necesario** de la siguiente jerarquía de clases para incluir **la interface sonoro** (esta contienen como mínimo un método **sonar**), haga que:

- la clase instrumento musical la implemente
- Su jerarquía de clases deberá tener:
  - 2 instrumentos de viento
  - 2 Instrumentos de cuerda
  - 2 instrumentos de percusión
- Genere una jerarquía alterna (similar a la de instrumentos musicales) de cosas que no sean instrumentos musicales y que suenen.

Utilice interfaces, clases abstractas, clases no abstractas y herencia para resolver este ejercicio. Realice en la función main un código donde muestre la aplicación de polimorfismo a su jerarquía de clases.

```
public interface InstrumentoMusical {
    void tocar();
    void afinar();
    String tipoInstrumento();
}

public class InstrumentoViento
    implements InstrumentoMusical{

    public void tocar(){
        System.out.println("Tocando "+tipoInstrumento());
    }
    public void afinar(){
        System.out.println("Afinando "+tipoInstrumento());
    }
    public String tipoInstrumento(){
        return "Instrumento de Viento";
    }
}

public class Flauta extends InstrumentoViento{
    @Override
    public String tipoInstrumento(){
        return "Flauta";
    }
}
```

```
public class Main {  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
        // TODO code application logic here  
        InstrumentoMusical instrumento;  
        instrumento = new InstrumentoViento();  
        instrumento.afinar();  
        instrumento.tocar();  
        System.out.println("Tipo de instrumento: "  
            +instrumento.tipoInstrumento());  
  
        instrumento = new Flauta();  
        instrumento.afinar();  
        instrumento.tocar();  
        System.out.println("Tipo de instrumento: "  
            +instrumento.  
            tipoInstrumento());  
    }  
}
```