

API Reflection

API Reflection

- Permite leer los metadatos de una clase y trabajar con ellos.
- Permite inspeccionar y manipular clases y objetos sin conocer a priori con que objetos estamos trabajando.
- El API Reflection lo utilizan la gran mayoría de los frameworks, como por ejemplo Hibernate o Spring.

API Reflection

- `public class` Ordenador {
- `private` String `id`;
- `private` String `descripcion`;
- `private int` `potencia`;
- `public` Ordenador(String `id`, String `descripcion`, `int` `potencia`) {
- `this.id` = `id`;
- `this.descripcion` = `descripcion`;
- `this.potencia` = `potencia`;
- }
- `//` getters y setters

API Reflection

```
public class Lavadora {  
    private String id;  
    private String modelo;  
    private String descripcion;  
    public Lavadora(String id, String modelo, String descripcion) {  
  
        this.id = id;  
        this.modelo = modelo;  
        this.descripcion = descripcion;  
    }  
  
    // getters y setters
```

API Reflection

Generamos un método “**imprimirListaCualquiera**” que accede al **API** de reflection y lee los metadatos de cada objeto.

API Reflection

```
public class Principal {  
    public static void main(String[] args) {  
        List<Ordenador> lista = new ArrayList<Ordenador>();  
        Ordenador o1 = new Ordenador("O1", "Ordenador gaming", 4);  
        Ordenador o2 = new Ordenador("O2", "Ordenador ofimatica", 2);  
        lista.add(o1);  
        lista.add(o2);  
        Lavadora l1= new Lavadora("L1", "Standard", "Lavadora normal");  
        Lavadora l2= new Lavadora("L2", "VIP", "Lavadora avanzada");  
        List<Lavadora> lista2= new ArrayList<Lavadora>();  
        lista2.add(l1);  
        lista2.add(l2);  
        imprimirListaCualquiera(lista);  
        imprimirListaCualquiera(lista2);  
    }  
}
```

API Reflection

Generamos un método “**imprimirListaCualquiera**” que accede al API de reflection y lee los metadatos de cada objeto.

1. Se utiliza un bucle para **recorrer el Array** y leer la lista **de métodos** que cada objeto contiene con `reflection` y `getMethods()`;
2. Se verifica que **el objeto dispone de los métodos `getId()` y `getDescripcion()`** usando `m.getName()`.
3. Se usa el método **`invoke`** del API de reflection para invocar el método que corresponda.

API Reflection

```
public static void imprimirListaCualquiera(List<?> lista) {
    for(Object c: lista) {
        Method[] metodos=c.getClass().getMethods();
        for(Method m: metodos) {
            // System.out.println(m.getName());
            if (m.getName().equals("getId") || m.getName().equals("getDescripcion"))
            {
                try {
                    String cadena = (String) m.invoke(c, null);
                    System.out.println(cadena);
                } catch (IllegalAccessException | IllegalArgumentException |
                InvocationTargetException e) {
                    e.printStackTrace();
                }
            }
        }
    }
}
```