

Laboratorio #4

Ejercicio 1: Timbre y Alarma.

```
public class Timbre {
    public void m1() {
        System.out.println("timbre 1");
    }

    public void m2() {
        System.out.println("timbre 2");
    }

    public String toString() {
        return "Ring...Ring";
    }
}

public class Alarma extends Timbre {
    public void m1() {
        System.out.println("Alarma 1");
    }
}
```

- a.
- a. "Ring...Ring".
 - b. Alarma 1
 - c. Timbre 2
- b. Al imprimir la nueva alarma en la segunda línea únicamente llama al método toString de la clase Madre, luego en el método m1, se sobre escribe el de la clase Madre, y únicamente se muestra el método m1 de la clase hija. Al tener esta herencia también imprimirá el método m2 de la clase padre.

Ejercicio 2: Timbre y Alarma (2nda ronda)

```
public class Timbre {
    public void m1() {
        System.out.println("timbre 1");
    }

    public void m2() {
        System.out.println("timbre 2");
    }

    public String toString() {
        return "Ring...Ring";
    }
}

public class Alarma extends Timbre {
    public void m1() {
        System.out.println("alarma 1");
    }

    public void m2() {
        super.m1();
    }

    public String toString() {
        return super.toString() + "..." +
            super.toString();
    }
}
```

- c.
- a. Alarma miRelej = new Alarma();
 - b. System.out.println(miRelej);
 - c. miRelej.m1();
 - d. miRelej.m2();
 - i. Ring...Ring...Ring...Ring
 - ii. Alarma 1
 - iii. Timbre 1
- d. Se declara el nuevo objeto tipo alarma, el cual pertenece a la clase Alarma, que tiene herencia de la clase timbre. En el primer comando se muestra en pantalla el método toString del objeto tipo alarma al sobre escribirse de la clase padre. De segundo se utiliza el método m1 de la clase hijo, el cual muestra "Alarma 1" en pantalla, luego el método m2

de la clase hijo, llama al método m1 de la clase por medio del super. Por lo que se muestra "Timbre 1".