Ejercicios propuestos JDeodorant:

Los ejercicios (1) y (2) se realizarán de forma manual, mientras que los ejercicios (3) y (4) se deberá utilizar cuando proceda la herramienta *JDeodorant*. Recordad, que también podéis utilizar en cualquier momento las refactorizaciones disponibles en eclipse.

1. ¿Qué sucede con el método visualizarEstado presentado en la página 2.? ¿Cómo quedaría su refactorización en esta nueva implementación?
2. Queremos añadir un nuevo estado entre Parando y Parado que sea FinTrayecto. ¿Qué cambios tendrías que realizar para incorporar este nuevo estado.? Realiza los cambios tanto en las clases como en el módulo de pruebas.
3. Dada la clase Figura del anexo, se pide (a) Crear un componente de pruebas para la clase Figura (únicamente casos válidos) y (b) realizar las refactorizaciones propuestas en este laboratorio de manera manual y utilizando *JDeodorant*. Explicar en un documento cuáles son los cambios que se han realizado de manera manual.
4. Utilizando la herramienta *JDeodorant* buscar un ejemplo de bad smell *Long method, God class* y *Feature envy*, realizar la refactorización con *JDeodorant*, y presentar en un documento el código inicial, el código refactorizado y una explicación indicando cuáles son los cambios que se han realizado.

Anexo: Código inicial de la clase Figura.

```
public class Figura {
 private double a;
 private double b;
 private double r;

 static final private int SQUARE = 0;
 static final private int RECTANGLE = 1;
 static final private int CIRCLE = 2;

 public double calculateArea(int shape) {
    double area = 0;
    switch(shape) {
       case SQUARE:
          area = a * a;
          break;
       case RECTANGLE:
          area = a * b;
          break;
       case CIRCLE:
          area = Math.PI * r * r;
          break;
    }
    return area;
 }

 public double calculatePerimeter(int shape) {
    double perimeter = 0;
    switch(shape) {
       case SQUARE:
          perimeter = 4 * a;
          break;
       case RECTANGLE:
          perimeter = 2 * (a + b);
          break;
       case CIRCLE:
          perimeter = 2 * Math.PI * r;
          break;
    }
    return perimeter;
 }

}
```