

ENTORNOS DE DESARROLLO

ANÁLISIS ESTÁTICO DE CÓDIGO ED4cd1103

Entornos de Desarrollo (ED) Unidad de Trabajo 11 Tarea 03

OBJETIVOS DE APRENDIZAJE

- Comprender la utilidad de los analizadores de código estático para mejorar la calidad del software.
 - Identificar posibles errores, malas prácticas o mejoras de estilo antes de la ejecución.
 - Aplicar cambios sencillos de refactorización con justificación técnica.
-

TA03 ANALIZA Y MEJORA UN CÓDIGO CON SONARLINT

Herramientas necesarias

- Editor de código (**IntelliJ IDEA** o **Visual Studio Code**)
 - Plugin **SonarLint** instalado y configurado
 - Repositorio Git
-

Instrucciones

1. Instala SonarLint en tu editor.
2. Crea un proyecto nuevo y copia el siguiente código **sin modificar**:

```
import java.util.ArrayList;
import java.util.List;

/**
 * Clase que simula un inventario simple de productos.
 * Permite agregar productos, calcular el valor total y mostrar un informe.
 */
public class GestorInventario {

    private List<Double> precios = new ArrayList<>();

    public void agregarProducto(double p) {
        if (p > 0) {
            precios.add(p);
        }
    }

    public double calcularValorTotal() {
        double v = 0;
    }
}
```

```
        for (int i = 0; i < precios.size(); i++) {
            v = v + precios.get(i);
        }
        return v;
    }

    public void mostrarInforme() {
        System.out.println("Número de productos: " + precios.size());
        System.out.println("Valor total: " + calcularValorTotal());
        if (precios.size() > 10) {
            System.out.println("Inventario grande. Revisar almacenamiento.");
        }
    }
}
```

3. Ejecuta **SonarLint** para analizar el código:

- Revisa los problemas detectados.
- Clasifica cuáles afectan a legibilidad, rendimiento o buenas prácticas.

4. Refactoriza tu código aplicando mejoras razonadas:

- Renombra variables con nombres más descriptivos.
- Sustituye números mágicos por constantes si corresponde.
- Optimiza estructuras (por ejemplo, usando **for-each** si procede).
- Mejora la legibilidad general.

5. Prepara un archivo .md donde incluyas:

- Una captura de pantalla de los issues que SonarLint detectó antes de refactorizar.
- Otra captura después de las mejoras.
- Una breve explicación de qué has aprendido.

Entrega

- Entrega un archivo **.zip** en aula virtual que conenga:
 - Tu repositorio local.
 - Tu informe en .md y pdf.
- Sincroniza tu proyecto en el repositorio de GitHub Classroom.

Guía de calificación

Nota	Descripción del desempeño
0	No se entrega la tarea, el código no funciona o no se ha usado SonarLint.
3	Se analiza parcialmente con SonarLint. No hay reflexión o la entrega es incompleta.

Nota	Descripción del desempeño
6	El código se mejora de forma aceptable, se corrigen los problemas principales detectados y se justifica en el informe.
9	Se corrigen todos los problemas, los commits están bien documentados y la reflexión es completa.
10	Se hace una aportación extra