

Sesión 14 Code Smells y Análisis Estático de Código con SonarQube y ESLint

Práctica: “Rescate Express: funciona... pero huele”

Objetivo

Vas a mejorar un código que **funciona**, pero está mal escrito. No vas a añadir funcionalidades: vas a **hacerlo más claro y mantenible**.

Material:

- Un archivo Java con un método “chungo”
- Un HTML + CSS con duplicaciones

Reglas:

- Puedes cambiar nombres, ordenar, separar funciones
- No puedes cambiar lo que hace el programa

PARTE A: JAVA (Detectar y arreglar Code Smells)

1) Código inicial Factura.java

```
public class Factura {  
  
    public void f(int a, int b, String c) {  
        int x = 0;  
  
        if (a > 0) {  
            x = a * b;  
        } else {  
            x = b;  
        }  
  
        if (c.equals("A")) {  
            x = x + 10;  
        } else if (c.equals("B")) {  
            x = x + 20;  
        }  
    }  
}
```

```
        System.out.println("Total: " + x);
    }
}
```

2) Tu misión (en este orden)

1. **Detecta 3 problemas** del código (code smells).
2. **Cambia nombres** (método y variables).
3. **Saca 2 métodos** (mínimo).
4. El programa debe seguir mostrando **el mismo total**.

Entrega: Factura.java refactorizado + respuestas cortas.

PARTE B — HTML/CSS (Eliminar duplicación)

1) HTML inicial (index.html)

```
<div class="box">
  <h1>Factura</h1>
  <p>Total</p>
</div>
```

```
<div class="box">
  <h1>Factura</h1>
  <p>Total</p>
</div>
```

2) CSS inicial (styles.css)

```
.box {
  border: 1px solid black;
  padding: 10px;
}
```

```
.box {
  border: 1px solid black;
  padding: 10px;
}
```

Tu misión

1. Explica por qué esto es un problema.
2. Deja el CSS **sin duplicación**.
3. Añade una mejora visual pequeña.

Entrega: index.html + styles.css corregidos.

BONUS

- Sustituye los números mágicos 10 y 20 por **constantes**.
- Haz que el recargo dependa de un enum en vez de Strings.