SINGLE RESPONSIBILITY PRINCIPLE

SOLID

Samuel Barato
Nicolas Orjuela

DEFINICIÓN

"UNA CLASE DEBE TENER UNA ÚNICA RAZÓN PARA CAMBIAR."

ESTE PRINCIPIO ESTABLECE QUE CADA CLASE, MÓDULO O FUNCIÓN DEBE ESTAR DEDICADA A UNA SOLA RESPONSABILIDAD. CUANDO UN COMPONENTE TIENE MÚLTIPLES PROPÓSITOS, CUALQUIER MODIFICACIÓN EN UNA DE SUS FUNCIONES PUEDE AFECTAR OTRAS PARTES DEL SISTEMA, INCREMENTANDO EL RIESGO DE ERRORES Y DIFICULTANDO EL MANTENIMIENTO.

BENEFICIOS DEL SRP

- CODIGO MÁS LIMPIO Y ORGANIZADO
- REUTLIZACIÓN EFICIENTE DEL CODIGO
- REDUCCIÓN DE LA DEPENDENCIA ENTRE MODÚLOS
- MAYOR ESTABILIDAD Y ADAPTIBILIDAD EN EL DESARROLLO DE SOFTWARE

VENTAJAS

- CÓDIGO MÁS ORGANIZADO Y COMPRENSIBLE
- FACILITA LA ESCALABILIDAD
- REDUCE EL ACOPLAMIENTO
- MEJORA LAS PRUEBAS UNITARIAS
- FOMENTA LA REUTILIZACIÓN

DESVENTAJAS

- MAYOR CANTIDAD DE CLASES Y ARCHIVOS
- DIFICULTAD EN PROYECTOS PEQUEÑOS
- CURVA DE APRENDIZAJE
- MAYOR ESFUERZO INICIAL

EJEMPLOS

NO USANDO SRP

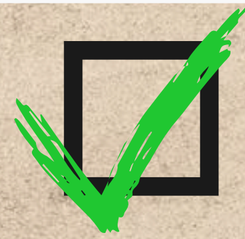
```java
class Report {
    public void generateReport() {
        // Lógica para generar el reporte
    }

    public void printReport() {
        // Lógica para imprimir el reporte
    }
}
```

USANDO SRP

```java
class ReportGenerator {
    public void generateReport() {
        // Lógica para generar el reporte
    }
}

class ReportPrinter {
    public void printReport() {
        // Lógica para imprimir el reporte
    }
}
```



CONCLUSIÓN

EL PRINCIPIO DE RESPONSABILIDAD ÚNICA (SRP) ES UNA DE LAS BASES DEL DISEÑO DE SOFTWARE BIEN ESTRUCTURADO Y MANTENIBLE. SU APLICACIÓN PERMITE CREAR SISTEMAS MÁS ORGANIZADOS, COMPRENSIBLES Y FÁCILES DE ESCALAR. AL DIVIDIR LAS RESPONSABILIDADES DE MANERA ADECUADA, REDUCIMOS EL ACOPLAMIENTO ENTRE LOS COMPONENTES, LO QUE FACILITA LA DETECCIÓN Y CORRECCIÓN DE ERRORES SIN AFECTAR OTRAS PARTES DEL SISTEMA

MUCHAS GRACIAS