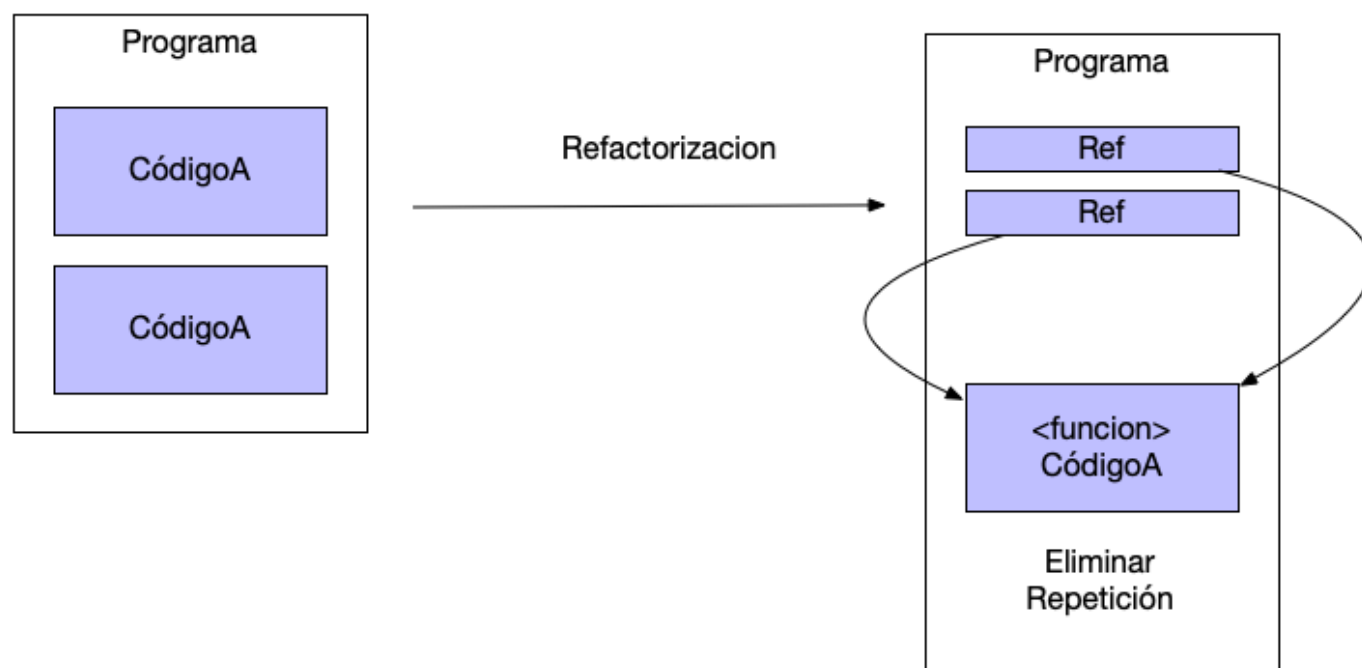


La regla del tres es una regla muy común cuando estamos trabajando con código y queremos refactorizar partes del código que sean idénticas o muy muy similares . Esta regla nos dice que si tenemos dos bloques de código similares no nos es imprescindible refactorizar. Vamos a explicarlo un poco más a detalle.

Código Repetido

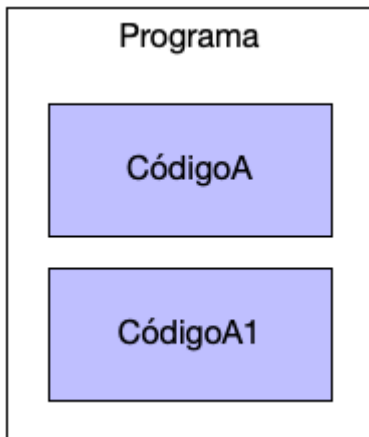
Imaginemonos que tenemos dos bloques de código idénticos en nuestro programa. ¿Debemos refactorizar? . Es una buena pregunta porque a veces uno abusa o se queda muy corto con las refactorizaciones. ¿Que sucede cuando tenemos código repetido “idéntico” ? Lo lógico es simplemente refactorizarlo. Ya que no cumple con el principio DRY salvo que sea minuscuro y no nos merezca la pena pero lo razonable sería usar un refactoring de eclipse como **extract method**.



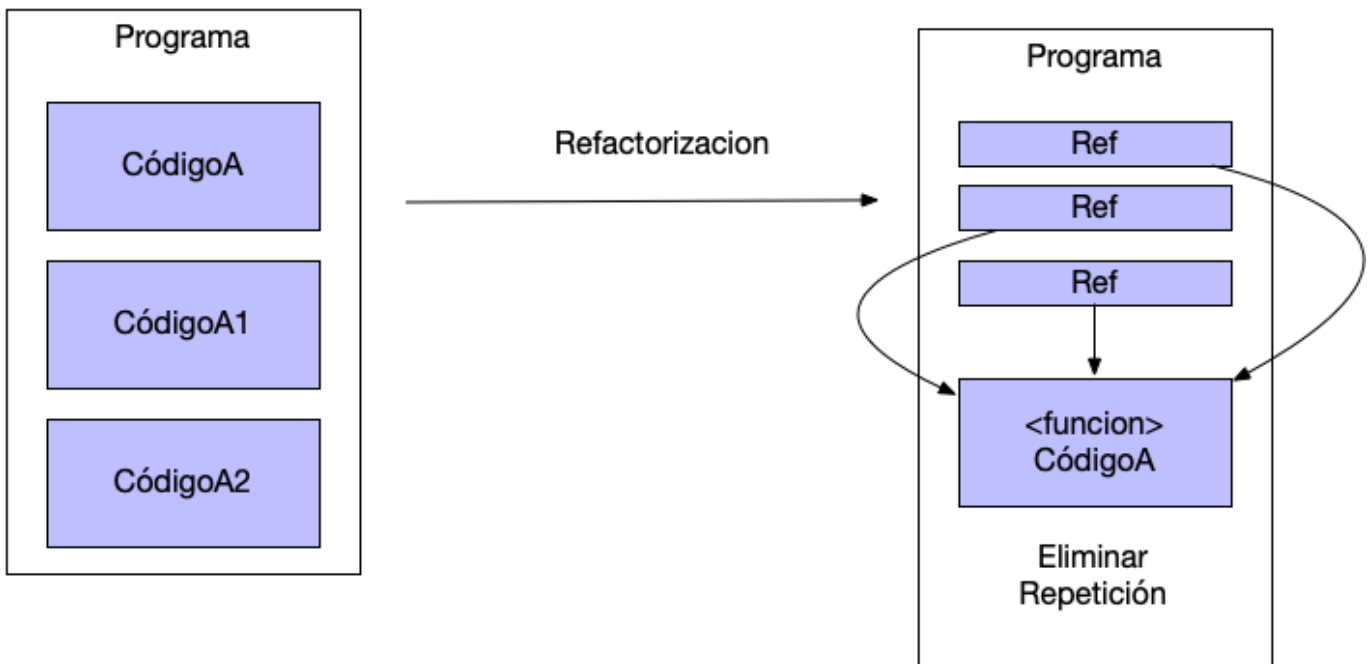
Código similar y Refactorización

La situacion cambia bastante cuando nosotros tenemos dos bloques de código que aunque no son idénticos son muy similares por ejemplo comparten el 90% del código. La regla del

tres nos indica que no es necesario refactorizar.



Por lo tanto nos quedaríamos como estábamos . Ahora bien que sucede cuando ese código es muy similar entre 3 bloques .



Es aquí en donde la regla del tres se aplicaría ya que nos encontramos en una situación en la cual es necesario reorganizar el código ya que probablemente la situación pueda repetirse en el futuro.

- Herencia y relaciones entre objetos

- [Java Collections List vs Set \(I\)](#)
- [Curso Diseño de Base de Datos y modelado](#)
- [JavaScript console time y rendimiento](#)
- [Java Try Catch y su manejo](#)
- [Refactoring](#)