**Inmutabilidad en Maps en Java**

1. Conceptos Básicos

Inmutabilidad: Un `Map` inmutable en Java es aquel que no puede ser modificado después de su creación. Esto significa que no se pueden agregar, eliminar o cambiar pares clavevalor.

Cómo crear un Map inmutable:

`Collections.unmodifiableMap(map)`: Devuelve una vista inmutable de un `Map` mutable.

`Map.of()`: Crea un `Map` inmutable directamente desde pares clavevalor.

`Map.copyOf(map)`: Crea un nuevo `Map` inmutable a partir de otro `Map`.

2. Ventajas de los Maps Inmutables

Seguridad en concurrencia: Evita la necesidad de sincronización en entornos multihilo.

Facilita el diseño: Permite un código más fácil de entender y mantener, al garantizar la consistencia del estado del `Map`.

Rendimiento: Reduce la necesidad de copias de datos y sincronización, mejorando el rendimiento en ciertos escenarios.

3. Cuándo Usar Maps Inmutables

Configuraciones: Ideal para datos de configuración que no deben cambiar.

Cache de datos: Garantiza la consistencia de los datos cacheados.

Constantes globales: Protege la integridad de datos constantes.

4. Fuentes

Documentación oficial de Java y artículos en línea, como los de Oracle y Baeldung.

1. <https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/Collections.html#unmodifiableMap(java.util.Map)>
2. <https://www.bing.com/ck/a?!&&p=22674cee2ac802b3JmltdHM9MTcyMzkzOTIwMCZpZ3VpZD0yNjYyNGE0Zi02ODZkLTY4NTYtMjZiNy01OWU3NjlmZDY5ODAmaW5zaWQ9NTIyNw&ptn=3&ver=2&hsh=3&fclid=26624a4f-686d-6856-26b7-59e769fd6980&psq=Baeldung%3a+Java+Immutable+Maps&u=a1aHR0cHM6Ly93d3cuYmFlbGR1bmcuY29tL2phdmEtaW1tdXRhYmxlLW1hcHM&ntb=1>