# Interfaz Map en Java

# **¿Qué es un Map?**

* Estructura que almacena **pares clave/valor**.
* No permite claves duplicadas.
* **No hereda** de Collection.

#### **Implementaciones principales**

1. HashMap: rápido, “sin orden”.
2. TreeMap: ordenado por claves.
3. LinkedHashMap: mantiene orden de inserción.

| **Implementación** | **Orden** | **Eficiencia búsqueda** | **Eficiencia inserción** | **Usa vistas** |
| --- | --- | --- | --- | --- |
| ***HashMap*** | Ninguno | Muy alta | Muy alta | ✅ |
| ***TreeMap*** | Orden natural | Alta | Alta | ✅ |
| ***LinkedHashMap*** | Inserción | Media | Alta | ✅ |

#### **Operaciones básicas**

| put(K clave, V valor) | Inserta o reemplaza |
| --- | --- |
| get(Object clave) | Devuelve valor asociado |
| remove(Object clave) | Elimina entrada por clave |
| containsKey(Object clave) | ¿Existe la clave? |
| containsValue(Object valor) | ¿Existe el valor? |
| clear() | Borra todo |

#### **Ejemplo**

Map<String, Double> m = new HashMap<>();

m.put("Ana", 1.65);

m.put("Luis", 1.73);

m.get("Ana"); → 1.65

#### **Vistas Collection del Map**

* keySet() → Set de claves
* values() → Collection de valores
* entrySet() → Set<Map.Entry<K,V>> de pares

#### **Iteración segura (ejemplo)**

for (Map.Entry<String, Double> e : m.entrySet()) {

System.out.println(e.getKey() + " → " + e.getValue());

}

#### **Notas**

* Solo se puede añadir con put().
* Las vistas permiten modificar el mapa original.