**2014 Big Java**

Contenido

[MAPS 1](#_Toc508433742)

# MAPS

* Un mapa nos permite asociar elementos de un conjunto de claves con elementos de una colección de valores.
* Utilizamos un mapa cuando queremos buscar objetos utilizando una clave (key). Por ejemplo, la siguiente figura muestra un mapa de nombres con personas y sus colores favoritos.

|  |
| --- |
|  |

* Así como hay dos tipos de implementaciones de **Set**, la librería de Java tiene dos implementaciones para la interface **Map**: **HashMap** y **TreeMap**.
* Antes de crear una HashMap o TreeMap, podemos almacenar su referencia en una referencia Map.

|  |
| --- |
| Map<String, Color> coloresFavoritos = **new** HashMap<>(); |

* Usar el método **put** para agregar una asociación.

|  |
| --- |
| coloresFavoritos.put("Juliet", Color.***RED***); |

* Podemos cambiar el valor de un valor de una asociación existente, esto se logra llamando nuevamente al método **put**.

|  |
| --- |
| coloresFavoritos.put("Juliet", Color.***BLUE***); |

* El método **get** retorna el valor asociado con una clave.

|  |
| --- |
| Color colorJuliet = coloresFavoritos.get("Juliet"); |

* Si preguntamos por una clave que no está asociada con ningún valor, entonces el método **get** retorna null.
* Para remover o eliminar una asociación, se llama al método **remove** con su clave.

|  |
| --- |
| coloresFavoritos.remove("Juliet"); |

* En ocasiones será necesario listar todas las claves contenidas en un mapa. El método **keySet** almacena el conjunto de claves. Podemos iterar el conjunto de claves y obtener todas las claves. De cada clave, podemos obtener el valor asociado con el método **get**. El siguiente código imprime los pares clave – valor del mapa coloresFavoritos.

|  |
| --- |
| **for** (String clave : coloresFavoritos.keySet()) {    Color valor = coloresFavoritos.get(clave);    System.***out***.println(clave + " - " + valor);    } |