



HashMap

Declaración

```
HashMap<Clave, Valor> miHashMap;
```

Ejemplos:

```
HashMap<Integer, String> misTextos;
```

```
HashMap<String, Objeto> misObjetos;
```

Inicialización

```
miHashMap = new HashMap<>();
```

Ejemplos:

```
misTextos = new HashMap<>();
```

```
misObjetos = new HashMap<>();
```

Añadir

```
miHashMap.put(Clave, Valor);
```

Ejemplos:

```
misTextos.put(3, “número tres”);
```

```
misObjetos.put(“Nombre del objeto”,  
variableConInstanciaDelObjeto);
```

Acceder

```
miHashMap.get(clave);
```

Ejemplos:

```
String texto = misTextos.get(3);
```

```
Objeto objeto = misObjetos.get("Nombre del  
objeto");
```

Tamaño

```
miHashMap.size();
```

Ejemplos:

```
int cuantosHay = misTextos.size();
```

```
int numeroElementos = misObjetos.size();
```

Mostrar todas las claves

```
for (int numero : misTextos.keySet()) {  
    System.out.println(numero);  
}
```

Mostrar todos los valores

```
for (int texto : misTextos.values()) {  
    System.out.println(texto);  
}
```


¿Cuándo usar un HashMap?

- En listas de muchos elementos recorrer la lista es muy costoso.
- Con HashMap podemos acceder, buscar, modificar o borrar un elemento de la lista **sin recorrerla**.