**Metodos de ArrayList() – Java**

|  |  |
| --- | --- |
| [**add()**](https://w3api.com/Java/ArrayList/add) | Permite añadir un elemento al final del [ArrayList](https://w3api.com/Java/ArrayList) o en una posicion especificada por parametros. |
| [**addAll()**](https://w3api.com/Java/ArrayList/addAll) | Elimina todos los elementos de un [ArrayList](https://w3api.com/Java/ArrayList). |
| [**clear()**](https://w3api.com/Java/ArrayList/clear) | Podemos realizar una copia de un [ArrayList](https://w3api.com/Java/ArrayList). |
| [**clone()**](https://w3api.com/Java/ArrayList/clone) | Mediante el método **clone** podemos realizar una copia de un [ArrayList](https://w3api.com/Java/ArrayList). |
| [**contains()**](https://w3api.com/Java/ArrayList/contains) | El método **contains** nos devuelve true en el caso de que el elemento pasado como parámetro se encuentra dentro del [ArrayList](https://w3api.com/Java/ArrayList). En caso contrario se devuelve false. |
| [**ensureCapacity()**](https://w3api.com/Java/ArrayList/ensureCapacity) | Incrementa la capacidad del [ArrayList](https://w3api.com/Java/ArrayList) y asegura que puede almancenar, al menos, la capacidad indicada como parámetro. |
| [**forEach()**](https://w3api.com/Java/ArrayList/forEach) | Podemos crear un consumidor, representado por la clase [Consumer](https://w3api.com/Java/Consumer) que se ejecute por cada elemento que hay en el [ArrayList](https://w3api.com/Java/ArrayList). Este consumidor es un **interface funcional** y por lo tanto acepta **funciones lambda** o una referencia a un método. |
| [**get()**](https://w3api.com/Java/ArrayList/get) | Devuelve un elemento del [ArrayList](https://w3api.com/Java/ArrayList) de la posición indicada como parámetro del método **get**. |
| [**indexOf()**](https://w3api.com/Java/ArrayList/indexOf) | Nos devuelve el índice que corresponde con la posición del [ArrayList](https://w3api.com/Java/ArrayList) del elemento que pasamos como parámetro. De esta forma el método **indexOf** nos permite buscar un elemento en el [ArrayList](https://w3api.com/Java/ArrayList). En el caso de que elemento no se encuentre dentro del [ArrayList](https://w3api.com/Java/ArrayList) se devolverá un valor de -1. |
| [**isEmpty()**](https://w3api.com/Java/ArrayList/isEmpty) | Método que comprueba que si el [ArrayList](https://w3api.com/Java/ArrayList) tiene elementos, en ese caso devolverá un false, o está vacío, en ese caso devolverá un true. |
| [**iterator()**](https://w3api.com/Java/ArrayList/iterator) | El método **iterator** nos devuelve un iterador sobre el [ArrayList](https://w3api.com/Java/ArrayList) que nor permitirá recorrerlo e iterar sobre todos sus elemnetos. |
| [**lastIndexOf()**](https://w3api.com/Java/ArrayList/lastIndexOf) | Al igual que el método [indexOf](https://w3api.com/Java/ArrayList/indexOf) nos buscaba la primera ocurrencia de un elemento dentro del [ArrayList](https://w3api.com/Java/ArrayList), el método **latIndexOf** nos identifica la última posición en la que se encuentra el elemento pasado por parámetro. En el caso de que el elemento se encuentre en el [ArrayList](https://w3api.com/Java/ArrayList) se devuelve el índice en el cual se encuentra, en el caso contrario se devuelve un valor de -1 que indicará que elemento no se encuentra en el [ArrayList](https://w3api.com/Java/ArrayList). |
| [**listIterator()**](https://w3api.com/Java/ArrayList/listIterator) | El método **listIterator** devuelve un iterador sobre los elementos del [ArrayList](https://w3api.com/Java/ArrayList) en la posición indicada como parámetro. En el caso que no indiquemos índice se devolverá un iterador a la primera posición. Y en el caso de que el índice pasado por parámetro no corresponda con una posición del [ArrayList](https://w3api.com/Java/ArrayList), es decir, que sea menor que cero o mayor que el tamaño del [ArrayList](https://w3api.com/Java/ArrayList), se devolverá una excepción [IndexOutOfBoundsException](https://w3api.com/Java/IndexOutOfBoundsException). |
| [**remove()**](https://w3api.com/Java/ArrayList/remove) | Elimina un elemento dentro de un [ArrayList](https://w3api.com/Java/ArrayList). Bien indicando la posición en la que se encuentra el elemento. O bien, por coincidencia con el objeto pasado como parámetro. En este caso se eliminará únicamente la primera ocurrencia que coincida con el elemento pasado como parámetro. |
| [**removeAll()**](https://w3api.com/Java/ArrayList/removeAll) | El método **removeAll** eliminan de un [ArrayList](https://w3api.com/Java/ArrayList) todos los elementos pasados en la colección que se indica en el parámetro. En el caso que se eliminen elementos de la lista se devolverá true, en caso contrario se devolverá false. |
| [**removeIf()**](https://w3api.com/Java/ArrayList/removeIf) | Con el método **removeIf** podemos eliminar todos los elementos de [ArrayList](https://w3api.com/Java/ArrayList) que coincidan con el predicado pasado como parámetro. Nos devolverá true si se eliminan elementos y false en el caso contrario. |
| [**removeRange()**](https://w3api.com/Java/ArrayList/removeRange) | Método que nos permite eliminar los elementos de un [ArrayList](https://w3api.com/Java/ArrayList) en un rango que delimitaremos con los índices indicados como parámetros del método. |
| [**retainAll()**](https://w3api.com/Java/ArrayList/retainAll) | El método **retainAll** nos permite eliminar todos los elementos de la lista que no le indiquemos en la colección que pasamos como parámetro al método. |
| [**set()**](https://w3api.com/Java/ArrayList/set) | Método que permite sustituir un elemento por otro dentro de un [ArrayList](https://w3api.com/Java/ArrayList). Así al método tenemos que indicarle la posición en la que queremos hacer la sustitución del elemento y el nuevo elemento. |
| [**size()**](https://w3api.com/Java/ArrayList/size) | Devuelve el tamaño, número de elementos, del [ArrayList](https://w3api.com/Java/ArrayList) |
| [**spliterator()**](https://w3api.com/Java/ArrayList/spliterator) | El método **spliterator** nos devuelve un [Spliterator](https://w3api.com/Java/Spliterator) sobre los elementos del [ArrayList](https://w3api.com/Java/ArrayList). De esa manera podemos procesarlos de forma paralela. |
| [**subList()**](https://w3api.com/Java/ArrayList/subList) | Mediante el método **subList** podemos extraer una porción de la lista actual, delimitada por los parámetros inicio y fin, para crear una nueva lista. |
| [**toArray()**](https://w3api.com/Java/ArrayList/toArray) | El método **toArray** nos permite convertir un [ArrayList](https://w3api.com/Java/ArrayList/toArray/Java/ArrayList) en un array con todos los elementos que cotiene el [ArrayList](https://w3api.com/Java/ArrayList/toArray/Java/ArrayList). El parámetro del método es el array sobre el que queremos convertir el [ArrayList](https://w3api.com/Java/ArrayList/toArray/Java/ArrayList). |
| [**trimToSize()**](https://w3api.com/Java/ArrayList/trimToSize) | El método **trimToSize** ajusta la capacidad del [ArrayList](https://w3api.com/Java/ArrayList) al número de elementos que este tiene. El [ArrayList](https://w3api.com/Java/ArrayList), al ser dinámico, siempre tiene preparado una capacida de 1,5 los elementos que tiene, para poder añadir nuevos elementos. Al utilizar **trimToSize** ajusta esta capacidad y ahorra espacio. |