



UNIVERSIDAD DEL BÍO-BÍO

# Paradigmas de la Programación

Colección de Objetos



UNIVERSIDAD DEL BÍO-BÍO

# Colección de objetos

Una colección de objetos es un objeto que puede almacenar un número variable de elementos siendo cada elemento otro objeto totalmente distinto.



# ArrayList<T>

La clase ArrayList en Java, es un tipo clase que permite almacenar datos en memoria de forma similar a los Arrays, con la ventaja de que el número de elementos que almacena, lo hace de forma dinámica, es decir, que no es necesario declarar su tamaño como pasa con los Arrays.

**Obs.:** En la imagen de abajo se crea una lista con capacidad indeterminada de tipo **String**.

```
ArrayList<String> textos = new ArrayList<String>();
```



# Utilizando ArrayList

```
ArrayList<String> textos = new ArrayList<String>();
textos.add("Hola Mundo");//Agregando elementos
textos.add(":)");
textos.add("Tengo sueño");
textos.add("¿Qué hora es?");

String texto = textos.get(3);//Obteniendo valor de la posición 3

for(int i=0; i < textos.size(); i++)//Imprimiendo cada valor.
    System.out.println(textos.get(i));
```

# ArrayList con datos primitivos

ArrayList trabaja solamente con declaraciones de Clases, pero no con datos primitivos(int,float,long,...).

Para poder manipular números por ejemplo, es necesario recurrir a la clase Wrapper asociada(Integer).

Las clases Wrapper son la versión en objetos de los datos primitivos.

- Byte para byte.
- Short para short.
- Integer para int.
- Long para long.
- Boolean para boolean
- Float para float.
- Double para double
- Character para char.

# ArrayList con datos primitivos

Si se desea tener una lista de números, se tendrá que utilizar su variante Integer y no int.

```
ArrayList<Integer> numeros = new ArrayList<Integer>();  
numeros.add(10); //Agregando elementos  
numeros.add(15);  
numeros.add(20);  
numeros.add(25);
```

# ArrayList – Métodos

MÉTODO	DESCRIPCIÓN
size()	Devuelve el número de elementos (int)
add(X)	Añade el objeto X al final. Devuelve true.
add(posición, X)	Inserta el objeto X en la posición indicada.
get(posicion)	Devuelve el elemento que está en la posición indicada.
remove(posicion)	Elimina el elemento que se encuentra en la posición indicada. Devuelve el elemento eliminado.
remove(X)	Elimina la primera ocurrencia del objeto X. Devuelve true si el elemento está en la lista.
clear()	Elimina todos los elementos.
set(posición, X)	Sustituye el elemento que se encuentra en la posición indicada por el objeto X. Devuelve el elemento sustituido.
contains(X)	Comprueba si la colección contiene al objeto X. Devuelve true o false.
indexOf(X)	Devuelve la posición del objeto X. Si no existe devuelve -1