

```

/*
 * La definición de la variable se hace así: ArrayList<TIPO_QUE_VAN_A_GUARDAR> NOMBRE_VARIABLE
 * Por ejemplo:
 */

ArrayList<Integer> miArreglo; // Este arreglo me va a permitir guardar Integers

/*
 * Para instanciar el arreglo, uso el constructor: new ArrayList<>();
 * Por ejemplo:
 */

miArreglo = new ArrayList<>(); // Esto construye un arreglo vacío: []

/*
 * Puedo agregar elementos a mi arreglo mediante el método: add(VALOR)
 * Esto permite agregar el valor a final del arreglo
 * Por ejemplo:
 */

miArreglo.add(2); // Arreglo resultante [2]
miArreglo.add(5); // Arreglo resultante [2,5]

int variable = 8;
miArreglo.add(variable); // Arreglo resultante [2,5,8]

/*
 * No puedo agregar valores de otro tipo de objeto al que declare cuando construí mi arreglo
 * Por ejemplo, cualquiera de las siguientes sentencias da error:
 */

miArreglo.add("hola");
miArreglo.add('a');
miArreglo.add(new Circulo("circulo",5));

/*
 * Puedo conocer la cantidad de elementos almacenados en el arreglo mediante el método: size()
 */

int cantidad = miArreglo.size(); // En cantidad se almacena el valor 3

/*
 * Puedo obtener los elementos guardados en mi arreglo mediante el método: get(POSICION)
 * Incluso puedo recorrerlo desde inicio a fin como un array normal.
 * Las posiciones arrancan en 0, al igual que en un array normal.
 * Por ejemplo:
 */

int primerValor = miArreglo.get(0); // Retorna el primer elemento del arreglo (el 2)
int segundoValor = miArreglo.get(1); // Retorna el segundo elemento del arreglo (el 5)
int ultimoValor = miArreglo.get(cantidad-1); // Retorna el ultimo elemento del arreglo (el 8)

// Este for recorre e imprime los valores del arreglo en orden
for (int indice = 0; indice < miArreglo.size(); indice++) {
    int valor = miArreglo.get(indice);
    System.out.println(valor);
}

/*
 * Por último, puedo borrar cualquier posicion del arreglo mediante el método: remove(POSICION)
 * Por ejemplo:
 */

miArreglo.remove(1); // Borra el segundo elemento del arreglo (el 5), dejando [2,8]
cantidad = miArreglo.size(); // El método size() ahora retorna cantidad 2

/*
 * Estos son algunos de los métodos básicos que ofrece el ArrayList, pero tiene muchos más. Los irán descubriendo
 * a medida que avancemos en la materia. Cuando quieran saber sobre algún método en particular nos pueden preguntar!
 */

/*
 * Otra particularidad, es que podemos hacer uso del polimorfismo dentro de los ArrayList.
 * Por ejemplo:
 */

ArrayList<Figura> arregloDeFiguras; // Este arreglo me va a permitir guardar Figuras, cualquier tipo de figuras!

arregloDeFiguras.add(new Circulo("circulo",3));
arregloDeFiguras.add(new Cuadrado("cuadrado",2));
arregloDeFiguras.add(new Triangulo("triangulo",4,5));

```