

Manual: Cuándo Usar Arreglo de Objetos vs ArrayList en Java

1. Introducción

En Java, trabajar con múltiples objetos puede hacerse usando **arreglos estáticos** o **ArrayList**. Este manual explica cuándo conviene utilizar cada enfoque, las ventajas y desventajas, y brinda ejemplos simples para entender los escenarios donde aplicarlos.

2. Escenarios de Uso

Cuándo usar un Arreglo Estático (Objeto sobre Objeto)

- **Cantidad fija de elementos:** Cuando se sabe desde el principio cuántos objetos se van a manejar.
- **Memoria controlada:** Es ideal si se busca evitar sobrecarga de memoria.
- **Ejemplo típico:** Lista de empleados para una empresa pequeña con cantidad fija.

Ventajas:

- Menor uso de recursos.
- Acceso más rápido con índices.

Desventajas:

- Tamaño fijo, no se puede cambiar en tiempo de ejecución.
- Difícil agregar o eliminar objetos.

Ejemplo con Arreglo Estático

```
package Ejecutar;

import Class.Persona;

public class EjecutarArreglo {
    public static void main(String[] args) {
        Persona[] personas = new Persona[2];
        personas[0] = new Persona("Carlos", "Pérez", 25);
        personas[1] = new Persona("Ana", "Gómez", 30);

        for (Persona persona : personas) {
            System.out.println(persona.toString());
        }
    }
}
```

Cuándo usar ArrayList

- **Cantidad variable de elementos:** Cuando no se conoce de antemano cuántos objetos se necesitarán.

- **Modificaciones dinámicas:** Si es necesario agregar o eliminar objetos durante la ejecución.
- **Ejemplo típico:** Lista de productos en un carrito de compras.

Ventajas:

- Tamaño dinámico.
- Métodos útiles como `add()`, `remove()`, `contains()`.

Desventajas:

- Mayor consumo de memoria.
- Acceso más lento en comparación con un arreglo.

Ejemplo con ArrayList

```
import java.util.ArrayList;
import Class.Persona;

public class EjecutarArrayList {
    public static void main(String[] args) {
        ArrayList<Persona> listaPersonas = new ArrayList<>();
        listaPersonas.add(new Persona("Carlos", "Pérez", 25));
        listaPersonas.add(new Persona("Ana", "Gómez", 30));
        listaPersonas.add(new Persona("Luis", "Rodríguez", 28));

        for (Persona persona : listaPersonas) {
            System.out.println(persona.toString());
        }
    }
}
```

3. Comparación Entre Arreglo Estático y ArrayList

Aspecto	Arreglo Estático	ArrayList
Tamaño	Fijo	Dinámico
Modificación	No se puede modificar	Se pueden añadir o eliminar elementos
Uso de memoria	Menor	Mayor
Rendimiento	Rápido	Relativamente más lento
Aplicación	Escenarios de tamaño fijo	Escenarios dinámicos

4. Manual: Objeto sobre Objeto en Java utilizando la Clase Persona

1. Clase Persona

```
package Class;
```

```

public class Persona {
    private String nombre;
    private String apellido;
    private int edad;

    public Persona(String nombre, String apellido, int edad) {
        this.nombre = nombre;
        this.apellido = apellido;
        this.edad = edad;
    }

    public String getNombre() { return nombre; }
    public void setNombre(String nombre) { this.nombre = nombre; }

    public String getApellido() { return apellido; }
    public void setApellido(String apellido) { this.apellido = apellido; }

    public int getEdad() { return edad; }
    public void setEdad(int edad) { this.edad = edad; }

    @Override
    public String toString() {
        return "Persona [Nombre=" + nombre + ", Apellido=" + apellido + ", Edad="
+ edad + "]\n";
    }
}

```

2. Ejecución Normal (Instancia Simple)

```

package Ejecutar;

import Class.Persona;

public class EjecutarNormal {
    public static void main(String[] args) {
        Persona persona1 = new Persona("Carlos", "Pérez", 25);
        System.out.println(persona1.toString());
    }
}

```

3. Ejecución con Objeto sobre Objeto (Arreglo de Objetos)

```

package Ejecutar;

import Class.Persona;

public class EjecutarObjetoSobreObjeto {
    public static void main(String[] args) {
        Persona[] personas = new Persona[3];
    }
}

```

```

    personas[0] = new Persona("Carlos", "Pérez", 25);
    personas[1] = new Persona("Ana", "Gómez", 30);
    personas[2] = new Persona("Luis", "Rodríguez", 28);

    for (int i = 0; i < personas.length; i++) {
        System.out.println("Persona " + (i + 1) + ": " +
            personas[i].toString());
    }
}

```

4. Conversión del Arreglo de Objetos

```

package Ejecutar;

import Class.Persona;

public class EjecutarObjetoSobreObjeto {
    public static void main(String[] args) {
        Persona[] personas = new Persona[3];
        personas[0] = new Persona("Carlos", "Pérez", 25);
        personas[1] = new Persona("Ana", "Gómez", 30);
        personas[2] = new Persona("Luis", "Rodríguez", 28);

        String[] nombres = extraerNombres(personas);

        System.out.println("Nombres extraídos:");
        for (String nombre : nombres) {
            System.out.println(nombre);
        }
    }

    public static String[] extraerNombres(Persona[] personas) {
        String[] nombres = new String[personas.length];
        for (int i = 0; i < personas.length; i++) {
            nombres[i] = personas[i].getNombre();
        }
        return nombres;
    }
}

```

5. Conclusión

Este manual mostró cómo trabajar con **instancias simples** y **objetos sobre objetos** en Java, permitiendo manejar múltiples objetos tanto con **arreglos estáticos** como con **ArrayList**. La elección depende del contexto y los requerimientos del proyecto.