

Java — Herencia, Interfaces y Colecciones



Este ejercicio esta pensado para **practicar paso a paso**:

- **Herencia + clases abstractas**
- **Interfaces**
- **Colecciones:** `ArrayList`, `HashSet`, `HashMap`

Incluye **tests unitarios (JUnit 5)** y **JaCoCo** para cobertura.

Además, el proyecto **genera una nota sobre 10** en función de los tests que vayan pasando, para que puedas ver como es tu evolución.

Estructura del proyecto

Código principal

- `src/main/java/com/docencia/herencia/ejercicio1..10`
- `src/main/java/com/docencia/interfaces/ejercicio1..10`
- `src/main/java/com/docencia/listas/ejercicio1..10`
- `src/main/java/com/docencia/sets/ejercicio1..10`
- `src/main/java/com/docencia/maps/ejercicio1..10`

En cada paquete `...ejercicioX` encontrarás:

- Clases del dominio (por ejemplo `Persona/Alumno/Profesor`, `Vehiculo/Coche/Motocicleta`, etc.)
- En ejercicios de colecciones: una clase "gestora" tipo `ListaX`, `ConjuntoX` o `MapaX`

Tests

- `src/test/java/com/docencia/herencia/ejercicio1..10`
 - `src/test/java/com/docencia/interfaces/ejercicio1..10`
 - `src/test/java/com/docencia/listas/ejercicio1..10`
 - `src/test/java/com/docencia/sets/ejercicio1..10`
 - `src/test/java/com/docencia/maps/ejercicio1..10`
 - `src/test/java/com/docencia/colecciones/ejercicio1..10` (*si existe*)
-

Orden recomendado para la resolución del ejercicio

1) Herencia (abstractas) — `com.docencia.herencia.ejercicio1..10`

Objetivo: entender atributos comunes en la clase base, constructores, sobrescritura y polimorfismo.

Sugerencia de enfoque:

- Implementar constructores y validaciones básicas
- Implementar métodos abstractos en hijas
- Implementar `equals/hashCode/toString`

Nota: Esto ya sabes hacerlo.

2) Interfaces — `com.docencia.interfaces.ejercicio1..10`

Objetivo: distinguir herencia vs contratos, `default methods`, y polimorfismo por interfaz. En este caso debes de implementar nuevamente lo realizado en el punto anterior e implementar los métodos en `las clases que implementan las interfaces`.

3) Listas (ArrayList) — `com.docencia.listas.ejercicio1..10`

Objetivo: `CRUD` (``Crear/Leer/Actualizar/Borrar)sobre listas:

- `anadir`
- `buscarPorId`
- `eliminarPorId`
- `modificar`
- `listar`
- `tamanio`

Reglas típicas son:

- No se permiten elementos `null`
- No se permiten campos `vacíos` (por ejemplo `nombre` en blanco)
- No se permiten IDs `null`
- No se permiten IDs duplicados

```

import java.util.ArrayList;
import java.util.List;
import java.util.NoSuchElementException;
import java.util.UUID;

/**
 * Gestiona una lista de {@link Vehiculo} usando {@link java.util.ArrayList}.
 *
 * Reglas:
 * - No se permiten elementos nulos.
 * - No se permiten vehículos con marca/modelo nulos o en blanco.
 * - No se permiten ids nulos ni duplicados dentro de la lista.
 */
public class ListaVehiculos {

    private final List<Vehiculo> vehiculos = new ArrayList<>();

    public void anadir(Vehiculo vehiculo) {
        throw new UnsupportedOperationException("El metodo no está implementado");
    }

    public Vehiculo buscarPorId(UUID id) {
        throw new UnsupportedOperationException("El metodo no está implementado");
    }

    public boolean eliminarPorId(UUID id) {
        throw new UnsupportedOperationException("El metodo no está implementado");
    }

    public void modificar(UUID id, Vehiculo nuevoVehiculo) {
        throw new UnsupportedOperationException("El metodo no está implementado");
    }

    public List<Vehiculo> listar() {
        throw new UnsupportedOperationException("El metodo no está implementado");
    }

    public int tamano() {
        throw new UnsupportedOperationException("El metodo no está implementado");
    }

    private boolean existeId(UUID id) {
        return vehiculos.stream().anyMatch(v -> v.getId().equals(id));
    }

    private void validar(Vehiculo vehiculo) {
        if (vehiculo == null) {
            throw new IllegalArgumentException("El vehículo no puede ser nulo");
        }
        if (vehiculo.getId() == null) {
            throw new IllegalArgumentException("El id no puede ser nulo");
        }
        if (vehiculo.getMarca() == null || vehiculo.getMarca().isBlank()) {
            throw new IllegalArgumentException("La marca no puede ser nula o en blanco");
        }
        if (vehiculo.getModelo() == null || vehiculo.getModelo().isBlank()) {
            throw new IllegalArgumentException("El modelo no puede ser nulo o en blanco");
        }
    }
}

```

4) Sets (HashSet) — [com.docencia.sets.ejercicio1..10](#)

Objetivo: unicidad (sin duplicados) y operaciones típicas del conjunto.

Observa: Mismos operaciones que en los ejercicios de [listas](#), y evitar nulos, etc.

5) Maps (HashMap) — [com.docencia.maps.ejercicio1..10](#)

Objetivo: búsquedas por clave ([UUID](#)) eficientes y operaciones típicas del mapa.

Observa: Mismos operaciones que en los ejercicios de [listas](#), y evitar nulos, etc.

Cómo ejecutar

Ejecutar tests

```
mvn clean test
```

Ejecutar verificación completa (incluye JaCoCo + nota)

```
mvn clean -Pcalificar verify
```

Cobertura en el comienzo del desarrollo

	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cov.	Missed	Lines	Missed	Methode	Missed	Classes
com_domenica_interfaces_ejercicio8	14 %	0 %	0 %	22 %	15 %	15 %	15 %	15 %	15 %	17 %	1 %	3 %
com_domenica_interfaces_ejercicio6	14 %	0 %	0 %	19 %	21 %	15 %	17 %	15 %	17 %	15 %	1 %	3 %
com_domenica_interfaces_ejercicio3	16 %	0 %	0 %	19 %	21 %	15 %	17 %	15 %	17 %	15 %	1 %	3 %
com_domenica_interfaces_ejercicio1	17 %	n/a	0 %	15 %	17 %	15 %	17 %	15 %	17 %	15 %	1 %	3 %
com_domenica_listas_ejercicio2	25 %	0 %	0 %	12 %	16 %	13 %	18 %	6 %	10 %	10 %	0 %	1 %
com_domenica_listas_ejercicio9	26 %	0 %	0 %	11 %	15 %	13 %	18 %	6 %	10 %	0 %	1 %	1 %
com_domenica_listas_ejercicio3	26 %	0 %	0 %	11 %	15 %	13 %	18 %	6 %	10 %	0 %	1 %	1 %
com_domenica_listas_ejercicio6	26 %	0 %	0 %	11 %	15 %	13 %	18 %	6 %	10 %	0 %	1 %	1 %
com_domenica_interfaces_ejercicio9	17 %	n/a	0 %	15 %	17 %	15 %	17 %	15 %	17 %	17 %	1 %	3 %
com_domenica_listas_ejercicio1	26 %	0 %	0 %	11 %	15 %	13 %	18 %	6 %	10 %	0 %	1 %	1 %
com_domenica_interfaces_ejercicio7	18 %	0 %	0 %	16 %	15 %	17 %	17 %	17 %	17 %	17 %	1 %	3 %
com_domenica_sets_ejercicio2	23 %	0 %	0 %	14 %	14 %	13 %	17 %	5 %	8 %	0 %	1 %	1 %
com_domenica_interfaces_ejercicio4	23 %	0 %	0 %	11 %	14 %	13 %	17 %	5 %	8 %	0 %	1 %	1 %
com_domenica_sets_ejercicio6	23 %	0 %	0 %	10 %	13 %	13 %	17 %	5 %	8 %	0 %	1 %	1 %
com_domenica_sets_ejercicio9	23 %	0 %	0 %	10 %	13 %	13 %	17 %	5 %	8 %	0 %	1 %	1 %
com_domenica_sets_ejercicio3	23 %	0 %	0 %	10 %	13 %	13 %	17 %	5 %	8 %	0 %	1 %	1 %
com_domenica_sets_ejercicio5	23 %	0 %	0 %	10 %	13 %	13 %	17 %	5 %	8 %	0 %	1 %	1 %
com_domenica_sets_ejercicio7	23 %	0 %	0 %	10 %	13 %	13 %	17 %	5 %	8 %	0 %	1 %	1 %
com_domenica_sets_ejercicio10	19 %	n/a	0 %	15 %	17 %	15 %	17 %	15 %	17 %	17 %	1 %	3 %
com_domenica_sets_ejercicio1	24 %	0 %	0 %	10 %	13 %	13 %	17 %	5 %	8 %	0 %	1 %	1 %
com_domenica_maquina_ejercicio1	24 %	0 %	0 %	10 %	13 %	13 %	17 %	5 %	8 %	0 %	1 %	1 %
com_domenica_maquina_ejercicio5	20 %	n/a	0 %	15 %	17 %	15 %	17 %	15 %	17 %	17 %	1 %	3 %
com_domenica_listas_ejercicio10	29 %	0 %	0 %	10 %	14 %	11 %	16 %	6 %	10 %	0 %	1 %	1 %
com_domenica_listas_ejercicio7	29 %	0 %	0 %	10 %	14 %	11 %	16 %	6 %	10 %	0 %	1 %	1 %
com_domenica_interfaces_ejercicio5	29 %	0 %	0 %	10 %	14 %	11 %	16 %	6 %	10 %	0 %	1 %	1 %
com_domenica_listas_ejercicio8	30 %	0 %	0 %	9 %	13 %	11 %	16 %	6 %	10 %	0 %	1 %	1 %
com_domenica_sets_ejercicio7	27 %	0 %	0 %	9 %	12 %	11 %	15 %	5 %	8 %	0 %	1 %	1 %
com_domenica_sets_ejercicio9	27 %	0 %	0 %	9 %	12 %	11 %	15 %	5 %	8 %	0 %	1 %	1 %
com_domenica_sets_ejercicio4	27 %	0 %	0 %	9 %	12 %	11 %	15 %	5 %	8 %	0 %	1 %	1 %
com_domenica_maquina_ejercicio10	27 %	0 %	0 %	9 %	12 %	11 %	15 %	5 %	8 %	0 %	1 %	1 %
com_domenica_maquina_ejercicio9	27 %	0 %	0 %	9 %	12 %	11 %	15 %	5 %	8 %	0 %	1 %	1 %
com_domenica_maquina_ejercicio4	27 %	0 %	0 %	9 %	12 %	11 %	15 %	5 %	8 %	0 %	1 %	1 %
com_domenica_maquina_ejercicio2	27 %	0 %	0 %	9 %	12 %	11 %	15 %	5 %	8 %	0 %	1 %	1 %
com_domenica_sets_ejercicio10	27 %	0 %	0 %	9 %	12 %	11 %	15 %	5 %	8 %	0 %	1 %	1 %
com_domenica_sets_ejercicio8	28 %	0 %	0 %	8 %	11 %	11 %	15 %	5 %	8 %	0 %	1 %	1 %
com_domenica_maquina_ejercicio8	28 %	0 %	0 %	8 %	11 %	11 %	15 %	5 %	8 %	0 %	1 %	1 %
com_domenica_herencia_ejercicio5	55 %	0 %	100 %	8 %	16 %	8 %	20 %	8 %	15 %	0 %	3 %	3 %
com_domenica_herencia_ejercicio3	63 %	0 %	100 %	7 %	16 %	7 %	21 %	7 %	15 %	0 %	3 %	3 %
com_domenica_herencia_ejercicio8	64 %	0 %	100 %	7 %	15 %	7 %	19 %	7 %	14 %	0 %	3 %	3 %
com_domenica_herencia_ejercicio2	68 %	0 %	100 %	7 %	16 %	7 %	21 %	7 %	15 %	0 %	3 %	3 %
com_domenica_herencia_ejercicio1	68 %	0 %	100 %	7 %	16 %	7 %	21 %	7 %	15 %	0 %	3 %	3 %
com_domenica_herencia_ejercicio9	68 %	0 %	100 %	7 %	16 %	7 %	21 %	7 %	15 %	0 %	3 %	3 %
com_domenica_herencia_ejercicio10	68 %	0 %	100 %	7 %	16 %	7 %	21 %	7 %	15 %	0 %	3 %	3 %
com_domenica_herencia_ejercicio19	65 %	0 %	100 %	7 %	15 %	7 %	19 %	7 %	14 %	0 %	3 %	3 %
com_domenica_herencia_ejercicio4	65 %	0 %	100 %	7 %	15 %	7 %	19 %	7 %	14 %	0 %	3 %	3 %
com_domenica_herencia_ejercicio7	65 %	0 %	100 %	7 %	15 %	7 %	19 %	7 %	14 %	0 %	3 %	3 %
Total	2,567 of 3,807	32 %	302 of 322	6 %	532	737	581	861	381	576	10 %	3 %

Cobertura al final el ejercicio

Element	Missed Instructions	Cov	Missed Branches	Cov	Missed	Cov	Missed	Lines	Missed	Methods	Missed	Classes
com.doenicia.sets.sets ejercicio2	82 %	65 %	14	27	12	52	1	8	0	1		
com.doenicia.sets.sets ejercicio3	81 %	66 %	13	26	12	52	1	8	0	1		
com.doenicia.sets.sets ejercicio4	81 %	66 %	13	26	12	52	1	8	0	1		
com.doenicia.sets.sets ejercicio5	81 %	66 %	13	26	12	52	1	8	0	1		
com.doenicia.sets.sets ejercicio1	81 %	66 %	13	26	12	52	1	8	0	1		
com.doenicia.sets.sets ejercicio6	83 %	67 %	12	25	11	50	1	8	0	1		
com.doenicia.sets.sets ejercicio7	83 %	67 %	12	25	11	50	1	8	0	1		
com.doenicia.sets.sets ejercicio8	83 %	67 %	12	25	11	50	1	8	0	1		
com.doenicia.sets.sets ejercicio9	83 %	67 %	12	25	11	50	1	8	0	1		
com.doenicia.sets.sets ejercicio10	83 %	67 %	12	25	11	50	1	8	0	1		
com.doenicia.sets.sets ejercicio11	83 %	68 %	11	24	11	50	1	8	0	1		
com.doenicia.maps.exercicio2	76 %	69 %	9	21	7	37	1	8	0	1		
com.doenicia.maps.exercicio3	78 %	70 %	8	20	7	37	1	8	0	1		
com.doenicia.maps.exercicio4	78 %	70 %	8	20	7	37	1	8	0	1		
com.doenicia.maps.exercicio5	78 %	70 %	8	20	7	37	1	8	0	1		
com.doenicia.maps.exercicio6	78 %	70 %	8	20	7	37	1	8	0	1		
com.doenicia.maps.exercicio7	80 %	72 %	7	19	6	35	1	8	0	1		
com.doenicia.maps.exercicio8	80 %	72 %	7	19	6	35	1	8	0	1		
com.doenicia.maps.exercicio9	80 %	72 %	7	19	6	35	1	8	0	1		
com.doenicia.maps.exercicio10	80 %	72 %	7	19	6	35	1	8	0	1		
com.doenicia.maps.exercicio11	80 %	72 %	7	19	6	35	1	8	0	1		
com.doenicia.maps.exercicio12	80 %	72 %	7	19	6	35	1	8	0	1		
com.doenicia.interfaces.exercicio3	91 %	58 %	11	19	1	31	1	17	1	1		
com.doenicia.interfaces.exercicio4	91 %	58 %	11	19	1	31	1	17	1	1		
com.doenicia.listas.exercicio2	93 %	80 %	5	25	2	35	0	12	0	1		
com.doenicia.listas.exercicio3	93 %	83 %	4	24	2	35	0	12	0	1		
com.doenicia.listas.exercicio4	93 %	83 %	4	24	2	35	0	12	0	1		
com.doenicia.listas.exercicio5	93 %	83 %	4	24	2	35	0	12	0	1		
com.doenicia.listas.exercicio6	93 %	83 %	4	24	2	35	0	12	0	1		
com.doenicia.listas.exercicio7	93 %	83 %	4	24	2	35	0	12	0	1		
com.doenicia.listas.exercicio8	93 %	83 %	4	24	2	35	0	12	0	1		
com.doenicia.listas.exercicio9	93 %	83 %	4	24	2	35	0	12	0	1		
com.doenicia.listas.exercicio10	93 %	83 %	4	24	2	35	0	12	0	1		
com.doenicia.listas.exercicio11	93 %	83 %	4	24	2	35	0	12	0	1		
com.doenicia.listas.exercicio12	93 %	83 %	4	24	2	35	0	12	0	1		
com.doenicia.interfaces.exercicio7	92 %	61 %	8	26	1	31	1	17	1	1		
com.doenicia.interfaces.exercicio8	92 %	61 %	8	26	1	31	1	17	1	1		
com.doenicia.interfaces.exercicio9	92 %	62 %	7	25	1	31	1	17	1	1		
com.doenicia.interfaces.exercicio10	92 %	62 %	7	25	1	31	1	17	1	1		
com.doenicia.interfaces.exercicio11	92 %	62 %	7	25	1	31	1	17	1	1		
com.doenicia.interfaces.exercicio12	92 %	62 %	7	25	1	31	1	17	1	1		
com.doenicia.interfaces.exercicio1	92 %	62 %	7	25	1	31	1	17	1	1		
com.doenicia.interfaces.exercicio2	92 %	62 %	7	25	1	31	1	17	1	1		
com.doenicia.interfaces.exercicio3	92 %	62 %	7	25	1	31	1	17	1	1		
com.doenicia.interfaces.exercicio4	92 %	62 %	7	25	1	31	1	17	1	1		
com.doenicia.interfaces.exercicio5	92 %	62 %	7	25	1	31	1	17	1	1		
com.doenicia.interfaces.exercicio6	92 %	62 %	7	25	1	31	1	17	1	1		
com.doenicia.interfaces.exercicio7	92 %	62 %	7	25	1	31	1	17	1	1		
com.doenicia.interfaces.exercicio8	92 %	62 %	7	25	1	31	1	17	1	1		
com.doenicia.interfaces.exercicio9	92 %	62 %	7	25	1	31	1	17	1	1		
com.doenicia.interfaces.exercicio10	92 %	62 %	7	25	1	31	1	17	1	1		
com.doenicia.interfaces.exercicio11	92 %	62 %	7	25	1	31	1	17	1	1		
com.doenicia.interfaces.exercicio12	92 %	62 %	7	25	1	31	1	17	1	1		
com.doenicia.herencia.exercicio3	98 %	75 %	2	19	0	26	0	15	0	5		
com.doenicia.herencia.exercicio4	98 %	75 %	2	19	0	26	0	15	0	5		
com.doenicia.herencia.exercicio5	98 %	75 %	2	19	0	26	0	15	0	5		
com.doenicia.herencia.exercicio6	98 %	75 %	2	19	0	26	0	15	0	5		
com.doenicia.herencia.exercicio7	98 %	75 %	2	19	0	26	0	15	0	5		
com.doenicia.herencia.exercicio8	98 %	75 %	2	19	0	26	0	15	0	5		
com.doenicia.herencia.exercicio9	98 %	75 %	2	19	0	26	0	15	0	5		
com.doenicia.herencia.exercicio10	98 %	75 %	2	19	0	26	0	15	0	5		
com.doenicia.herencia.exercicio11	98 %	75 %	2	19	0	26	0	15	0	5		
com.doenicia.herencia.exercicio12	98 %	75 %	2	19	0	26	0	15	0	5		
Total	865 of 7.732	88 %	311 of 1.082	71 %	341	1.137	205	1.772	30	596	10	9

Auto-cálculo de nota (progreso)

El proyecto genera un fichero:

- target/nota.txt

La nota se calcula **proporcionalmente** a los tests que pasan:

```
notaSobre10 = 10 * (testsPasados / testsTotales)
```

Esto permite que veas tu evolución:

- Al principio muchos tests fallan (nota baja) ****1.14****
 - Conforme implementa métodos, pasan más tests y la nota sube

- Si pasan todos, la nota llega a **10.00**

Algo que debes de recordar

- se ejecute el cálculo de nota **aunque fallen tests**
 - se genere **target/nota.txt** en cada **mvn clean -Pcalificar verify**
-

Notas a tener en cuenta

Durante el inicio, es normal que existan métodos sin implementar que lancen:

```
throw new UnsupportedOperationException("El metodo no esta implementado");
```

porque son de las cosas que debes de implementar.

Esto te permite:

- Compilar el proyecto desde el minuto 1
 - Ir implementando por fases, descritas al inicio del documento
 - Ver cómo sube la nota conforme pasan más tests
-

Flujo de trabajo (alumno/a)

1. Ejecuta **mvn clean -Pcalificar verify**
 2. Abre el reporte de fallos y resuelve por orden:
 - constructores y validación
 - métodos de negocio
 - **equals/hashCode/toString**
 - colecciones (**anadir**, **buscarPorId**, **eliminarPorId**, **modificar**)
 3. Ejecuta **mvn verify** para ver:
 - cobertura (JaCoCo)
 - nota (target/nota.txt)
-