

# GUIA AMPLIADA CON EJEMPLOS DE CODIGO

## 1. METODO BUSCAR (EXAMEN 2024, DAWBANK, VIDEO)

```
public int buscarPorDNI(String dni) {
    for(int i = 0; i < lista.length; i++) {
        if(lista[i] != null && lista[i].getDni().equals(dni)) {
            return i;
        }
    }
    return -1;
}
```

## 2. REGISTRAR OBJETO EN ARRAY (USADO EN TODOS LOS EJERCICIOS)

```
public void registrar(Trabajador t) {

    int pos = buscarPorDNI(t.getDni());
    if(pos != -1) {
        System.out.println("Ya existe");
        return;
    }

    for(int i = 0; i < lista.length; i++) {
        if(lista[i] == null) {
            lista[i] = t;
            System.out.println("Registrado");
            return;
        }
    }

    System.out.println("Sin espacio");
}
```

## 3. ELIMINAR OBJETO (ARRAYS ESTATICOS)

```
public void eliminar(String dni) {
    int pos = buscarPorDNI(dni);

    if(pos == -1) {
        System.out.println("No existe");
        return;
    }

    lista[pos] = null;
}
```

## 4. MOSTRAR INFORMACION (POO BASICO)

```
public void mostrarInfo() {
    System.out.println("Nombre: " + nombre);
    System.out.println("DNI: " + dni);
    System.out.println("Direccion: " + direccion);
}
```

## 5. ARRAY DE OBJETOS + MOSTRAR TODOS

```
public void mostrarTodos() {
    for(int i = 0; i < lista.length; i++) {
        if(lista[i] != null) {
            lista[i].mostrarInfo();
        }
    }
}
```

## 6. HERENCIA: CLASE BASE + HIJA (EJERCICIO ASTROS)

```
public abstract class Astro {

    protected double masa;
    protected double diametro;

    public Astro(double masa, double diametro) {
        this.masa = masa;
        this.diametro = diametro;
    }

    public abstract void mostrar();
}

public class Planeta extends Astro {

    private Satelite[] satelites;
    private int contador;

    public Planeta(double masa, double diametro, int maxSat) {
        super(masa, diametro);
        satelites = new Satelite[maxSat];
        contador = 0;
    }

    @Override
    public void mostrar() {
        System.out.println("Planeta con masa: " + masa);
    }
}
```

## 7. HERENCIA MASCOTAS (EJERCICIO COMPLETO)

```
public abstract class Mascota {

    protected String nombre;

    public Mascota(String nombre) {
        this.nombre = nombre;
    }

    public abstract void hablar();
}

public class Perro extends Mascota {

    public Perro(String nombre) {
        super(nombre);
    }

    @Override
```

```

        public void hablar() {
            System.out.println("Guau");
        }
    }
}

```

## 8. MENU DEL EXAMEN (ESTRUCTURA GENERAL)

```

int opcion;

do {
    mostrarMenu();
    opcion = sc.nextInt();

    switch(opcion) {
        case 1: registrar(); break;
        case 2: eliminar(); break;
        case 3: mostrarTodos(); break;
        case 4: buscar(); break;
        case 0: System.out.println("Fin"); break;
    }

} while(opcion != 0);

```

## 9. EJEMPLO REAL COMBINADO (EXAMEN 2024)

```

public void alquilarPelicula(String dni, int cod) {

    int posCli = buscarCliente(dni);
    if(posCli == -1) {
        System.out.println("No existe cliente");
        return;
    }

    int posPel = buscarPelicula(cod);
    if(posPel == -1) {
        System.out.println("No existe película");
        return;
    }

    if(peliculas[posPel].isAlquilada()) {
        System.out.println("Ya está alquilada");
        return;
    }

    peliculas[posPel].setAlquilada(true);
    System.out.println("Pelicula alquilada exitosamente");
}

```