

# **Algoritmos y Estructura de Datos – TI**

**Guía de Problemas - Semana 09**

# Clase ArrayList: declaración privada, creación e inicialización, métodos básicos del ArrayList, operaciones básicas y complementarias.

## Problema\_09\_1

Dada la implementación de la clase **Alumno** en el paquete **clase**

```
package clase;

public class Alumno {

    // Atributos privados
    private int codigo, nota1, nota2;
    private String nombre;
    // Constructor
    public Alumno(int codigo, String nombre, int nota1, int nota2) {
        this.codigo = codigo;
        this.nombre = nombre;
        this.nota1 = nota1;
        this.nota2 = nota2;
    }
    // Métodos de acceso público: set/get
    public void setCodigo(int codigo) {
        this.codigo = codigo;
    }
    public void setNombre(String nombre) {
        this.nombre = nombre;
    }
    public void setNota1(int nota1) {
        this.nota1 = nota1;
    }
    public void setNota2(int nota2) {
        this.nota2 = nota2;
    }
    public int getCodigo() {
        return codigo;
    }
    public String getNombre() {
        return nombre;
    }
    public int getNota1() {
        return nota1;
    }
    public int getNota2() {
        return nota2;
    }
    // Operaciones públicas
    public double promedio() {
        return (nota1 + nota2) / 2.0;
    }
}
```

Dada la implementación de la clase **ArregloAlumnos** en el paquete **arreglo**

```
package arreglo;

import clase.Alumno;

import java.util.ArrayList;

public class ArregloAlumnos {

    // Atributo privado
    private ArrayList <Alumno> alu;
    // Constructor
    public ArregloAlumnos() {
        alu = new ArrayList <Alumno> ();
        adicionar(new Alumno(123, "Ana", 19, 15));
        adicionar(new Alumno(456, "Juan", 15, 14));
        adicionar(new Alumno(789, "Pedro", 17, 19));
        adicionar(new Alumno(302, "David", 13, 18));
        adicionar(new Alumno(417, "Carlos", 20, 19));
        adicionar(new Alumno(641, "Jorge", 12, 13));
        adicionar(new Alumno(208, "María", 15, 17));
        adicionar(new Alumno(820, "José", 11, 10));
    }
    // Operaciones públicas básicas
    public void adicionar(Alumno x) {
        alu.add(x);
    }
    public int tamaño() {
        return alu.size();
    }
    public Alumno obtener(int i) {
        return alu.get(i);
    }
    // Operaciones públicas complementarias
    public int cantAprobados() {
        return 0;
    }
}
```

Agregue en esta última la operación complementaria que retorne:

- Cantidad de alumnos aprobados (promedio mayor o igual a 13).
- Cantidad de alumnos desaprobados (promedio menor a 13).
- Nombre del primer alumno desaprobado. En caso no exista retorne null.
- Nombre del último alumno aprobado. En caso no exista retorne null.
- Nombre del segundo alumno desaprobado. En caso no exista retorne null.
- Nombre del penúltimo alumno aprobado. En caso no exista retorne null.

En la clase principal:

- Considere la declaración global `ArregloAlumnos aa = new ArregloAlumnos();`
- Implemente la pulsación del botón **Adicionar** que agregue un nuevo alumno.
- Implemente la pulsación del botón **Reportar** que muestre los retornos de los métodos complementarios.

## Problema\_09\_2

Implemente la clase **Docente** en el paquete **clase** con los atributos privados: código (*int*), nombre (*String*), horas (*int*) y tarifa (*double*); un constructor que inicialice los atributos privados, los métodos de acceso público set/get y el método sueldo (horas \* tarifa).

Implemente la clase **ArregloDocentes** en el paquete **arreglo** con el atributo privado **doc** (ArrayList de tipo Docente) e implemente:

### Métodos básicos

- Un constructor que cree el ArrayList y adicione las DirMem de ocho objetos Docente.
- Método adicionar que recibe la DirMem de un nuevo docente y lo adicione al ArrayList.
- Método tamaño que retorne la cantidad de objetos Docente registrados hasta ese momento.
- Método obtener que reciba una posición y retorne la DirMem del objeto Docente respectivo.

### Métodos complementarios

- Retorne el sueldo promedio.
- Retorne el sueldo mayor.
- Retorne el sueldo menor.
- Retorne la tarifa mayor.
- Retorne la tarifa menor.

En la clase principal:

- Considere la declaración global ArregloDocentes **ad** = new ArregloDocentes();
- Implemente la pulsación del botón **Adicionar** que agregue un nuevo docente.
- Implemente la pulsación del botón **Reportar** que muestre los retornos de los métodos complementarios.

## Propuesto\_09\_1

Implemente la clase **Factura** en el paquete **clase** con los atributos privados: ruc (*String*), empresa (*String*), unidades (*int*) y precio unitario (*double*); un constructor que inicialice los atributos privados, los métodos de acceso público set/get y el método importeFacturado (unidades \* precio unitario).

Implemente la clase **ArregloFacturas** en el paquete **arreglo** con el atributo privado **fac** (ArrayList de tipo Factura) e implemente:

### Métodos básicos

- Un constructor que cree el ArrayList y adicione las DirMem de ocho objetos Factura.
- Método adicionar que recibe la DirMem de una nueva factura y la adiciona al ArrayList.
- Método tamaño que retorne la cantidad de objetos Factura registrados hasta ese momento.
- Método obtener que reciba una posición y retorne la DirMem del objeto Factura respectivo.

### Métodos complementarios

- Retorne suma de todos los importes facturados.
- Retorne importe promedio facturado.
- Retorne menor importe facturado.
- Retorne mayor importe facturado.
- Retorne nombre de la primera empresa cuyo importe facturado sea menor al importe promedio facturado.

En la clase principal:

- Considere la declaración global ArregloFacturas **af** = new ArregloFacturas();
- Implemente la pulsación del botón **Adicionar** que agregue una nueva factura.
- Implemente la pulsación del botón **Reportar** que muestre los retornos de los métodos complementarios.

*"El don de la virtud es la joya más valiosa que Dios puede darle al hombre."*

**MP**